TIME SERIES K-MEANS

UN NUOVO ALGORITMO DI CLUSTERING PER SERIE TEMPORALI

Elia Mercatanti
Relatore: *Donatella Merlini*

Università degli Studi di Firenze
Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Anno Accademico 2016-2017

Indice

Che cosa si intende per Clustering?

Definizione

Il "**Clustering**" o la "**Cluster Analysis**" è un particolare insieme di tecniche di data mining con il compito di selezionare e raggruppare elementi omogenei in gruppi (**clusters**), dove le somiglianze fra gli oggetti di uno stesso gruppo sono massimizzate e le somiglianze tra oggetti appartenenti a gruppi diversi sono minimizzate.

Applicazioni del Clustering

Applicazioni del Clustering

- **Comprensione dei dati**
  Identificare le classi di appartenenza dei dati all'interno di un set.

Examples

Raggruppare documenti correlati ad una ricerca web o azioni con un andamento simile del prezzo.

Applicazioni del Clustering

- **Comprensione dei dati**
  Identificare le classi di appartenenza dei dati all'interno di un set.

- **Riassunto dei dati**
  Ricercare i prototipi più rappresentativi dei cluster.

Examples

Raggruppare documenti correlati ad una ricerca web o azioni con un andamento simile del prezzo.

Examples

Summarization: ridurre un set di dati in un suo riassunto.
Compression: ridurre la dimensione di allocazione di un set di dati.

Il Concetto di Cluster

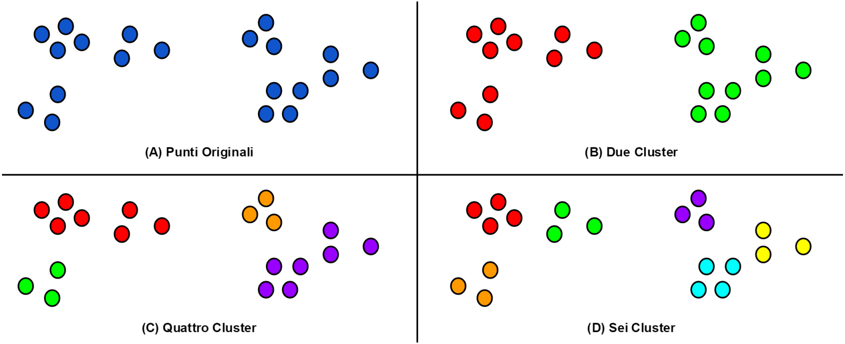
Il Concetto di Cluster

- La Cluster Analysis raggruppa gli oggetti basandosi solo sulle informazioni trovate nei dati che li descrivono e che specificano le loro relazioni.

Il Concetto di Cluster

- La Cluster Analysis raggruppa gli oggetti basandosi solo sulle informazioni trovate nei dati che li descrivono e che specificano le loro relazioni.
- In generale, la nozione di cluster non è sempre ben definita, la sua migliore descrizione dipende dalla natura dei dati e dal tipo di risultati che vogliamo ottenere.

L'ambiguità della Nozione di Cluster



Modi diversi di raggruppare lo stesso set di punti (clustering).

Unsupervised Classification e Supervised Classification

La Cluster Analysis può essere confrontata con altre tecniche che suddividono oggetti in gruppi o che li assegnano a delle classi.

Unsupervised Classification e Supervised Classification

La Cluster Analysis può essere confrontata con altre tecniche che suddividono oggetti in gruppi o che li assegnano a delle classi.

- **Unsupervised Classification**
  Assegnano i dati ai cluster utilizzando solo le informazioni contenute nel set di dati.

Examples

Tecniche di Clustering o della Cluster Analysis.

Unsupervised Classification e Supervised Classification

La Cluster Analysis può essere confrontata con altre tecniche che suddividono oggetti in gruppi o che li assegnano a delle classi.

- **Unsupervised Classification**
  Assegnano i dati ai cluster utilizzando solo le informazioni contenute nel set di dati.
- **Supervised Classification**
  Assegnano le classi in base ad un modello sviluppato a partire da oggetti la cui classe è nota.

Examples

Tecniche di Clustering o della Cluster Analysis.

Examples

Tecniche di Classificazione.

Clustering Partizionale e Gerarchico

Esistono varie classificazioni delle tecniche di clustering. La più comune tiene conto se il set di cluster generato è annidato o meno.

Clustering Partizionale e Gerarchico

Esistono varie classificazioni delle tecniche di clustering. La più comune tiene conto se il set di cluster generato è annidato o meno.

- **Clustering Partizionale**: si basa sulla divisione del set di dati in sottoinsiemi non sovrapposti (clusters) in modo tale che ogni oggetto si trovi in un unico sottoinsieme.

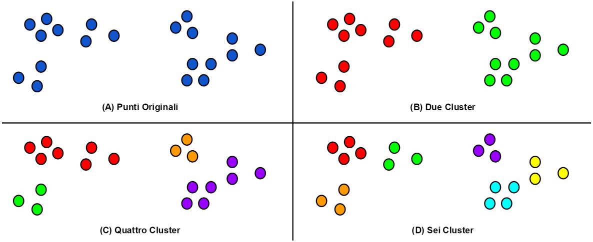
Clustering Partizionale e Gerarchico

Esistono varie classificazioni delle tecniche di clustering. La più comune tiene conto se il set di cluster generato è annidato o meno.

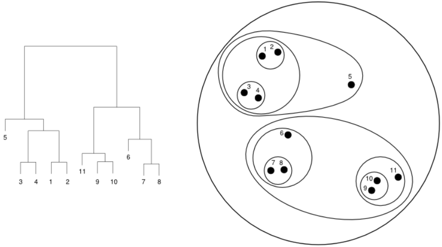
- **Clustering Partizionale**: si basa sulla divisione del set di dati in sottoinsiemi non sovrapposti (clusters) in modo tale che ogni oggetto si trovi in un unico sottoinsieme.
- **Clustering Gerarchico**: permette ai cluster di avere delle gerarchie di partizioni (sotto-cluster), ovvero, un set di cluster annidati avente una struttura ad albero.
  - **Agglomerativo**: quando la strategia è di tipo *bottom up*.
  - **Divisivo**: quando la strategia è di tipo *top down*.

Clustering Partizionale e Gerarchico

Clustering Partizionale



Clustering Gerarchico



Altri Tipi di Clustering

Altri Tipi di Clustering

- **Clustering Esclusivo**: ogni oggetto viene assegnato ad uno ed a un solo cluster.

Altri Tipi di Clustering

- **Clustering Esclusivo**: ogni oggetto viene assegnato ad uno ed a un solo cluster.
- **Clustering Sovrapposto**: ogni oggetto può simultaneamente appartenere a più di un cluster.

Altri Tipi di Clustering

- **Clustering Esclusivo**: ogni oggetto viene assegnato ad uno ed a un solo cluster.
- **Clustering Sovrapposto**: ogni oggetto può simultaneamente appartenere a più di un cluster.
- **Fuzzy Clustering**: ogni oggetto appartiene ad ogni cluster con un'appartenenza pesata che varia da 0 a 1.

Altri Tipi di Clustering

- **Clustering Esclusivo**: ogni oggetto viene assegnato ad uno ed a un solo cluster.
- **Clustering Sovrapposto**: ogni oggetto può simultaneamente appartenere a più di un cluster.
- **Fuzzy Clustering**: ogni oggetto appartiene ad ogni cluster con un'appartenenza pesata che varia da 0 a 1.
- **Clustering Completo**: ogni oggetto del set di dati viene assegnato ad un cluster.

Altri Tipi di Clustering

- **Clustering Esclusivo**: ogni oggetto viene assegnato ad uno ed a un solo cluster.
- **Clustering Sovrapposto**: ogni oggetto può simultaneamente appartenere a più di un cluster.
- **Fuzzy Clustering**: ogni oggetto appartiene ad ogni cluster con un'appartenenza pesata che varia da 0 a 1.
- **Clustering Completo**: ogni oggetto del set di dati viene assegnato ad un cluster.
- **Clustering Parziale**: solo alcuni oggetti del set di dati vengono assegnati ad un cluster.

Tipi Differenti di Cluster

Tipi Differenti di Cluster

- **Cluster Ben Separati**: set di oggetti in cui ogni elemento è più vicino ad ogni altro elemento del cluster rispetto ad oggetti al di fuori del gruppo.

Tipi Differenti di Cluster

- **Cluster Ben Separati**: set di oggetti in cui ogni elemento è più vicino ad ogni altro elemento del cluster rispetto ad oggetti al di fuori del gruppo.
- **Cluster basati su Prototipi**: set di oggetti in cui ogni elemento è più simile ad un prototipo che definisce il cluster rispetto ai prototipi che definiscono gli altri gruppi.

Tipi Differenti di Cluster

- **Cluster Ben Separati**: set di oggetti in cui ogni elemento è più vicino ad ogni altro elemento del cluster rispetto ad oggetti al di fuori del gruppo.
- **Cluster basati su Prototipi**: set di oggetti in cui ogni elemento è più simile ad un prototipo che definisce il cluster rispetto ai prototipi che definiscono gli altri gruppi.
- **Cluster basati su Grafi**: se abbiamo un grafo come set, un cluster è un set di oggetti che sono collegati fra di loro ma che non hanno connessioni con gli oggetti al di fuori del gruppo.

Tipi Differenti di Cluster

- **Cluster Ben Separati**: set di oggetti in cui ogni elemento è più vicino ad ogni altro elemento del cluster rispetto ad oggetti al di fuori del gruppo.
- **Cluster basati su Prototipi**: set di oggetti in cui ogni elemento è più simile ad un prototipo che definisce il cluster rispetto ai prototipi che definiscono gli altri gruppi.
- **Cluster basati su Grafi**: se abbiamo un grafo come set, un cluster è un set di oggetti che sono collegati fra di loro ma che non hanno connessioni con gli oggetti al di fuori del gruppo.
- **Cluster basati sulla Densità**: set con una densa regione di oggetti circondata da una regione con bassa densità.

Tipi Differenti di Cluster

- **Cluster Ben Separati**: set di oggetti in cui ogni elemento è più vicino ad ogni altro elemento del cluster rispetto ad oggetti al di fuori del gruppo.
- **Cluster basati su Prototipi**: set di oggetti in cui ogni elemento è più simile ad un prototipo che definisce il cluster rispetto ai prototipi che definiscono gli altri gruppi.
- **Cluster basati su Grafi**: se abbiamo un grafo come set, un cluster è un set di oggetti che sono collegati fra di loro ma che non hanno connessioni con gli oggetti al di fuori del gruppo.
- **Cluster basati sulla Densità**: set con una densa regione di oggetti circondata da una regione con bassa densità.
- **Cluster Concettuali**: set di oggetti che condividono una qualche proprietà.

L'Algoritmo di Clustering K-means

L'Algoritmo di Clustering K-means

- Tecnica di clustering partizionale basata su prototipi.

L'Algoritmo di Clustering K-means

- Tecnica di clustering partizionale basata su prototipi.
- Suddivide un insieme di oggetti in K gruppi (cluster) sulla base dei loro attributi. Dove K viene scelto dall'utente.

L'Algoritmo di Clustering K-means

- Tecnica di clustering partizionale basata su prototipi.
- Suddivide un insieme di oggetti in K gruppi (cluster) sulla base dei loro attributi. Dove K viene scelto dall'utente.
- Il prototipo di un cluster viene definito tramite un **centroide**, che in genere rappresenta la media del gruppo di oggetti del cluster (punti, serie temporali, ecc.).

L'Algoritmo di Clustering K-means

- Tecnica di clustering partizionale basata su prototipi.
- Suddivide un insieme di oggetti in K gruppi (cluster) sulla base dei loro attributi. Dove K viene scelto dall'utente.
- Il prototipo di un cluster viene definito tramite un **centroide**, che in genere rappresenta la media del gruppo di oggetti del cluster (punti, serie temporali, ecc.).
- Ogni oggetto del set di dati viene assegnato al cluster con il centroide più vicino.

L'Algoritmo di Clustering K-means

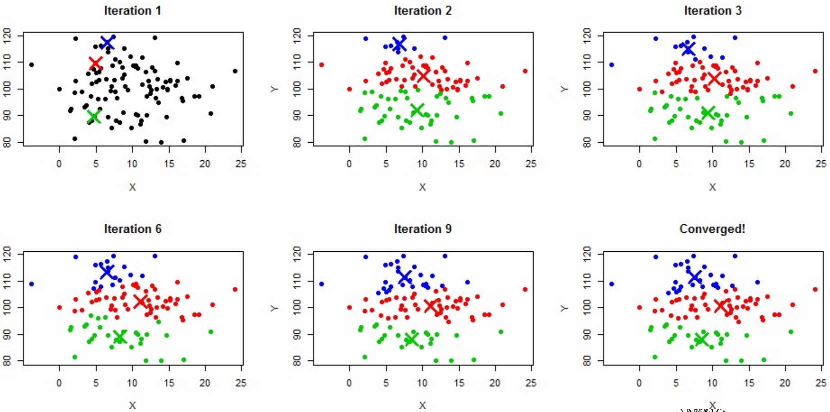
- Tecnica di clustering partizionale basata su prototipi.
- Suddivide un insieme di oggetti in K gruppi (cluster) sulla base dei loro attributi. Dove K viene scelto dall'utente.
- Il prototipo di un cluster viene definito tramite un **centroide**, che in genere rappresenta la media del gruppo di oggetti del cluster (punti, serie temporali, ecc.).
- Ogni oggetto del set di dati viene assegnato al cluster con il centroide più vicino.
- L'algoritmo di base è molto semplice.

L'Algoritmo di Clustering K-means

Algorithm 1: K-means di base

1 Seleziona K punti come centroidi iniziali.
2 **repeat**
3   Forma K cluster assegnando ogni punto al centroide più vicino.
4   Ricalcola i centroidi di ogni cluster.
5 **until** *Centroidi non cambiano.*

Esempio di Esecuzione



Uso del K-means per trovare tre cluster in un set di punti.

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

- **Distanza Euclidea**: $\sqrt{\sum_{i=1}^{n}(X_i - C_i)^2}$

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

- **Distanza Euclidea**: $\sqrt{\sum_{i=1}^{n}(X_i - C_i)^2}$
- **Distanza Euclidea Quadratica**: $\sum_{i=1}^{n}(X_i - C_i)^2$

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

- **Distanza Euclidea**: $\sqrt{\sum_{i=1}^{n}(X_i - C_i)^2}$
- **Distanza Euclidea Quadratica**: $\sum_{i=1}^{n}(X_i - C_i)^2$
- **Distanza di Manhattan**: $\sum_{i=1}^{n}|X_i - C_i|$

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

- **Distanza Euclidea**: $\sqrt{\sum_{i=1}^{n}(X_i - C_i)^2}$
- **Distanza Euclidea Quadratica**: $\sum_{i=1}^{n}(X_i - C_i)^2$
- **Distanza di Manhattan**: $\sum_{i=1}^{n}|X_i - C_i|$
- **Correlazione di Pearson**: $\frac{\sum_{i=1}^{n}(X_i-\overline{X})(C_i-\overline{C})}{\sqrt{\sum_{i=1}^{n}(X_i-\overline{X})^2}\sqrt{\sum_{i=1}^{n}(C_i-\overline{C})^2}}$

  dove $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ e $\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$

Misure di Prossimità

Per assegnare un punto al centroide più vicino, abbiamo bisogno di una misura di prossimità (similarità) che quantifichi il concetto di "vicino" per il tipo di dati che stiamo considerando. Alcuni esempi:

- **Distanza Euclidea**: $\sqrt{\sum_{i=1}^{n}(X_i - C_i)^2}$
- **Distanza Euclidea Quadratica**: $\sum_{i=1}^{n}(X_i - C_i)^2$
- **Distanza di Manhattan**: $\sum_{i=1}^{n}|X_i - C_i|$
- **Correlazione di Pearson**: $\frac{\sum_{i=1}^{n}(X_i - \overline{X})(C_i - \overline{C})}{\sqrt{\sum_{i=1}^{n}(X_i - \overline{X})^2}\sqrt{\sum_{i=1}^{n}(C_i - \overline{C})^2}}$
  dove $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ e $\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$
- **Similarità del Coseno**: $1 - \frac{\sum_{i=1}^{n} X_i C_i}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} C_i^2}}$

Dettagli dell'Algoritmo K-means

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.
- Nel caso di dati nello spazio Euclideo, come funzione obiettivo scegliamo l'**SSE** $= \sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.
- Nel caso di dati nello spazio Euclideo, come funzione obiettivo scegliamo l'**SSE** $= \sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.
- I centroidi iniziali sono spesso scelti in modo causale.

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.
- Nel caso di dati nello spazio Euclideo, come funzione obiettivo scegliamo l'**SSE** $= \sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.
- I centroidi iniziali sono spesso scelti in modo causale.
- Le misure di prossimità descritte precedentemente fanno sempre convergere il K-means.

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.
- Nel caso di dati nello spazio Euclideo, come funzione obiettivo scegliamo l'**SSE** $= \sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.
- I centroidi iniziali sono spesso scelti in modo causale.
- Le misure di prossimità descritte precedentemente fanno sempre convergere il K-means.
- La convergenza in genere avviene durante le prime iterazioni. Spesso dunque l'esecuzione viene portata avanti fino a che solo l'1% degli oggetti cambia cluster.

Dettagli dell'Algoritmo K-means

- I centroidi possono variare a seconda della misura di prossimità scelta e in base all'obiettivo del clustering. Tale scopo è tipicamente espresso da una funzione obiettivo.
- Nel caso di dati nello spazio Euclideo, come funzione obiettivo scegliamo l'**SSE** $= \sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.
- I centroidi iniziali sono spesso scelti in modo causale.
- Le misure di prossimità descritte precedentemente fanno sempre convergere il K-means.
- La convergenza in genere avviene durante le prime iterazioni. Spesso dunque l'esecuzione viene portata avanti fino a che solo l'1% degli oggetti cambia cluster.
- La complessità in tempo è pari a $O((m+K)n)$. Quella in tempo è pari a $O(I * K * m * n)$.

Debolezze e Punti di Forza del K-means

Debolezze

Punti di Forza

Debolezze e Punti di Forza del K-means

Debolezze

- Può generare cluster vuoti.
- I dati anomali possono influenzare i cluster negativamente.
- Non gestisce correttamente cluster non globulari o con differenti dimensioni e densità.
- È ristretto a quei dati che hanno una nozione di centro (centroide).

Punti di Forza

Debolezze e Punti di Forza del K-means

Debolezze

- Può generare cluster vuoti.
- I dati anomali possono influenzare i cluster negativamente.
- Non gestisce correttamente cluster non globulari o con differenti dimensioni e densità.
- È ristretto a quei dati che hanno una nozione di centro (centroide).

Punti di Forza

- La sua estrema semplicità.
- Grande versatilità d'uso su molteplici tipi di dati.
- Molto efficiente in termini computazionali.

Le Serie Temporali

Definizione

Una **serie temporale** è una sequenza di osservazioni ordinate rispetto al tempo, che in genere esprime la dinamica di un certo fenomeno o parametro nel tempo.

Le Serie Temporali

Definizione

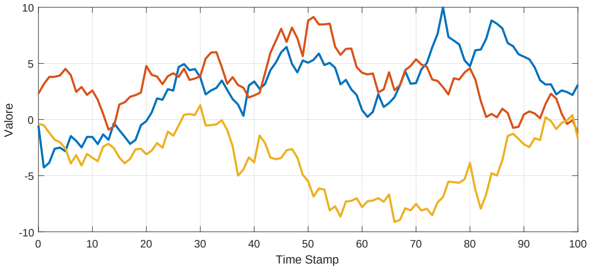
Una **serie temporale** è una sequenza di osservazioni ordinate rispetto al tempo, che in genere esprime la dinamica di un certo fenomeno o parametro nel tempo.

- Vengono studiate sia per interpretare un fenomeno, individuando componenti di trend, ciclicità, stagionalità e accidentalità, sia per prevederne il suo andamento futuro.

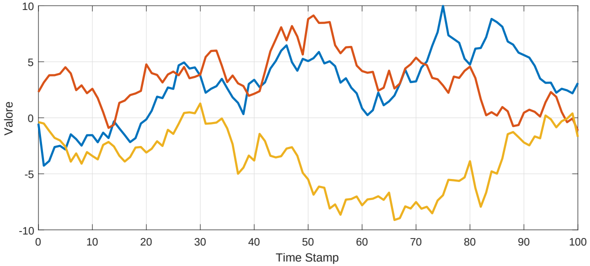
Examples

L'intensità del traffico su una strada nell'arco di un anno oppure l'andamento mensile del prezzo di un determinato prodotto.

Caratteristiche delle Serie Temporali

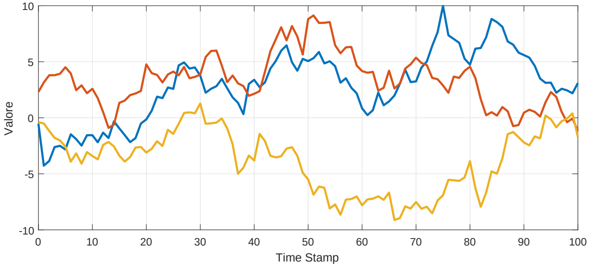


Caratteristiche delle Serie Temporali



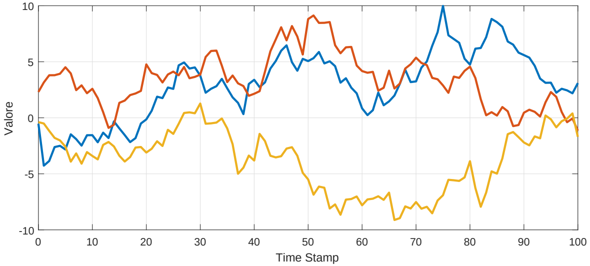
- Sono utilizzate in aree che spaziano dalla scienza, all'ingegneria, all'economia e alla medicina.

Caratteristiche delle Serie Temporali



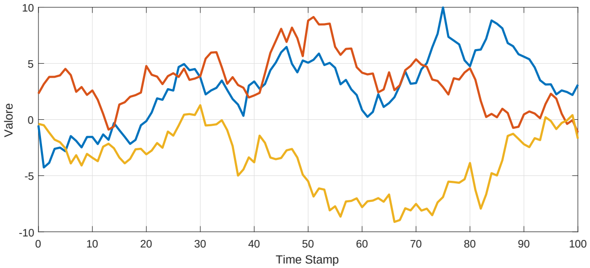
- Sono utilizzate in aree che spaziano dalla scienza, all'ingegneria, all'economia e alla medicina.
- Richiedono continui aggiornamenti per rappresentare in modo corretto il fenomeno che descrivono.

Caratteristiche delle Serie Temporali



- Sono utilizzate in aree che spaziano dalla scienza, all'ingegneria, all'economia e alla medicina.
- Richiedono continui aggiornamenti per rappresentare in modo corretto il fenomeno che descrivono.
- Hanno per loro natura grandi dimensioni.

Caratteristiche delle Serie Temporali



- Sono utilizzate in aree che spaziano dalla scienza, all'ingegneria, all'economia e alla medicina.
- Richiedono continui aggiornamenti per rappresentare in modo corretto il fenomeno che descrivono.
- Hanno per loro natura grandi dimensioni.
- Fanno spesso parte di set di dati molto grandi.

Clustering per Serie Temporali

Dato un set di n serie temporali $S = \{S_1, S_2, \ldots, S_n\}$, il processo di partizionamento non supervisionato di S in un insieme di cluster $C = \{C_1, C_2, \ldots, C_k\}$, creato in modo tale che serie temporali omogenee vengano raggruppate insieme basandosi su una certa misura di similarit, viene chiamato clustering per serie temporali.

Clustering per Serie Temporali

Dato un set di n serie temporali $S = \{S_1, S_2, \ldots, S_n\}$, il processo di partizionamento non supervisionato di S in un insieme di cluster $C = \{C_1, C_2, \ldots, C_k\}$, creato in modo tale che serie temporali omogenee vengano raggruppate insieme basandosi su una certa misura di similarit, viene chiamato clustering per serie temporali.

Caratteristiche:

- Impegnativo in termini di spazio occupato, per via della grandezza dei set di dati basati su serie temporali.

Clustering per Serie Temporali

Dato un set di n serie temporali $S = \{S_1, S_2, \ldots, S_n\}$, il processo di partizionamento non supervisionato di S in un insieme di cluster $C = \{C_1, C_2, \ldots, C_k\}$, creato in modo tale che serie temporali omogenee vengano raggruppate insieme basandosi su una certa misura di similarit, viene chiamato clustering per serie temporali.

Caratteristiche:

- Impegnativo in termini di spazio occupato, per via della grandezza dei set di dati basati su serie temporali.
- Impegnativo dal punti di vista computazionale, per via del grande numero di osservazioni che caratterizzano le serie temporali.

Clustering per Serie Temporali

Dato un set di n serie temporali $S = \{S_1, S_2, \ldots, S_n\}$, il processo di partizionamento non supervisionato di S in un insieme di cluster $C = \{C_1, C_2, \ldots, C_k\}$, creato in modo tale che serie temporali omogenee vengano raggruppate insieme basandosi su una certa misura di similarit, viene chiamato clustering per serie temporali.

Caratteristiche:

- Impegnativo in termini di spazio occupato, per via della grandezza dei set di dati basati su serie temporali.
- Impegnativo dal punti di vista computazionale, per via del grande numero di osservazioni che caratterizzano le serie temporali.
- Confrontare serie temporali sulla loro intera sequenza risulta molto pesante.

Applicazioni del Clustering per Serie Temporali

Applicazioni del Clustering per Serie Temporali

- **Ricerca di anomalie, novità o rilevamenti di discordanza**: sono utilizzati per scoprire pattern insoliti o inaspettati che si verificano in set di dati in modo sorprendente.

Applicazioni del Clustering per Serie Temporali

- **Ricerca di anomalie, novità o rilevamenti di discordanza**: sono utilizzati per scoprire pattern insoliti o inaspettati che si verificano in set di dati in modo sorprendente.
- **Riconoscimento di cambiamenti dinamici**: ad esempio l'individuazione di una correlazione nell'andamento di alcune serie temporali.

Applicazioni del Clustering per Serie Temporali

- **Ricerca di anomalie, novità o rilevamenti di discordanza**: sono utilizzati per scoprire pattern insoliti o inaspettati che si verificano in set di dati in modo sorprendente.
- **Riconoscimento di cambiamenti dinamici**: ad esempio l'individuazione di una correlazione nell'andamento di alcune serie temporali.
- **Predizioni e consigli**: alcune tecniche ibride che combinano il clustering all'approssimazione di funzioni possono aiutare l'utente a predire alcuni eventi rilevanti.

Applicazioni del Clustering per Serie Temporali

- **Ricerca di anomalie, novità o rilevamenti di discordanza**: sono utilizzati per scoprire pattern insoliti o inaspettati che si verificano in set di dati in modo sorprendente.
- **Riconoscimento di cambiamenti dinamici**: ad esempio l'individuazione di una correlazione nell'andamento di alcune serie temporali.
- **Predizioni e consigli**: alcune tecniche ibride che combinano il clustering all'approssimazione di funzioni possono aiutare l'utente a predire alcuni eventi rilevanti.
- **Scoperta di modelli**: ad esempio per ricercare i gruppi di pattern più interessanti nei database.

Classificazione del Clustering per Serie Temporali

Classificazione del Clustering per Serie Temporali

- **Whole Sequence Clustering**: scopre le serie temporali che hanno pattern simili rispetto all'intera sequenza su cui sono definite, raggruppandole in cluster differenti a seconda della loro somiglianza basata su una misura di similarità.

Classificazione del Clustering per Serie Temporali

- **Whole Sequence Clustering**: scopre le serie temporali che hanno pattern simili rispetto all'intera sequenza su cui sono definite, raggruppandole in cluster differenti a seconda della loro somiglianza basata su una misura di similarità.
- **Subsequence Clustering**: cerca di identificare differenti intervalli di tempo (**sottospazi**), ovvero singoli segmenti, delle intere sequenze su cui sono definite le serie temporali, per poi eseguire gli algoritmi di clustering rispetto a questi intervalli.

Classificazione del Clustering per Serie Temporali

- **Whole Sequence Clustering**: scopre le serie temporali che hanno pattern simili rispetto all'intera sequenza su cui sono definite, raggruppandole in cluster differenti a seconda della loro somiglianza basata su una misura di similarità.
- **Subsequence Clustering**: cerca di identificare differenti intervalli di tempo (**sottospazi**), ovvero singoli segmenti, delle intere sequenze su cui sono definite le serie temporali, per poi eseguire gli algoritmi di clustering rispetto a questi intervalli.
- **Time Point Clustering**: genera dei cluster in base ad una combinazione tra la prossimità dei punti (coppie istante di tempo - valore) rispetto al tempo e alla similarità dei loro corrispondenti valori. Viene applicato ad una singola serie temporale per la ricerca di cluster di punti temporali.

Misure di Prossimità per Serie Temporali

Uno dei modi più semplici e utilizzati per calcolare la distanza fra due serie temporali consiste nel considerarle "univariate", e calcolare la distanza attraverso i punti temporali.

Definizione

Una serie temporale **univariata** consiste in una sequenza di numeri reali raccolti ad intervalli di tempo regolari.

Misure di Prossimità per Serie Temporali

Uno dei modi più semplici e utilizzati per calcolare la distanza fra due serie temporali consiste nel considerarle "univariate", e calcolare la distanza attraverso i punti temporali.

Definizione

Una serie temporale **univariata** consiste in una sequenza di numeri reali raccolti ad intervalli di tempo regolari.

- **Prossimità Rispetto al Tempo**: la similarità fra serie temporali viene calcolata comparando i valori che assumono in ogni istante di tempo.

Misure di Prossimità per Serie Temporali

Uno dei modi più semplici e utilizzati per calcolare la distanza fra due serie temporali consiste nel considerarle "univariate", e calcolare la distanza attraverso i punti temporali.

Definizione

Una serie temporale **univariata** consiste in una sequenza di numeri reali raccolti ad intervalli di tempo regolari.

- **Prossimità Rispetto al Tempo**: la similarità fra serie temporali viene calcolata comparando i valori che assumono in ogni istante di tempo.
- **Prossimità Rispetto alla Forma**: la similarità fra serie temporali viene calcolata considerando la loro forma a prescindere dai punti temporali.

Dynamic Time Warping

Il **Dynamic Time Warping**, o DTW, è un algoritmo che permette di trovare una corrispondenza ottima tra due sequenze, ad esempio due serie temporali, attraverso una distorsione non lineare rispetto al tempo e che può portare ad una misura di prossimità rispetto alla forma tra le due sequenze allineate.

Dynamic Time Warping

Il **Dynamic Time Warping**, o DTW, è un algoritmo che permette di trovare una corrispondenza ottima tra due sequenze, ad esempio due serie temporali, attraverso una distorsione non lineare rispetto al tempo e che può portare ad una misura di prossimità rispetto alla forma tra le due sequenze allineate.

Caratteristiche:

- Utile per trattare sequenze in cui singole componenti hanno caratteristiche che variano nel tempo.

Dynamic Time Warping

Il **Dynamic Time Warping**, o DTW, è un algoritmo che permette di trovare una corrispondenza ottima tra due sequenze, ad esempio due serie temporali, attraverso una distorsione non lineare rispetto al tempo e che può portare ad una misura di prossimità rispetto alla forma tra le due sequenze allineate.

Caratteristiche:

- Utile per trattare sequenze in cui singole componenti hanno caratteristiche che variano nel tempo.
- È utilizzato in diversi campi di applicazione, dal riconoscimento vocale alla cluster analysis.

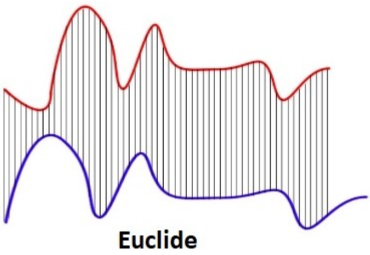
Dynamic Time Warping

Il **Dynamic Time Warping**, o DTW, è un algoritmo che permette di trovare una corrispondenza ottima tra due sequenze, ad esempio due serie temporali, attraverso una distorsione non lineare rispetto al tempo e che può portare ad una misura di prossimità rispetto alla forma tra le due sequenze allineate.

Caratteristiche:

- Utile per trattare sequenze in cui singole componenti hanno caratteristiche che variano nel tempo.
- È utilizzato in diversi campi di applicazione, dal riconoscimento vocale alla cluster analysis.
- Risulta molto costoso in termini computazionali per la sua complessità quadratica rispetto alla lunghezza delle serie temporali.

Distanza Euclidea e DTW



Differenza fra la distanza Euclidea e la distanza DTW calcolata su due serie temporali.

Algoritmi di Clustering per Serie Temporali

Algoritmi di Clustering per Serie Temporali

- **Clustering Gerarchico**: genera gerarchie annidate di gruppi simili basati su matrici di distanza fra serie temporali. I cluster risultano di bassa qualità. Non richiede come parametro iniziale il numero di cluster. Gestisce serie temporali di lunghezze differenti attraverso l'utilizzo di misure come il DTW ma non di grandi dimensioni per via della complessità quadratica.

Algoritmi di Clustering per Serie Temporali

- **Clustering Gerarchico**: genera gerarchie annidate di gruppi simili basati su matrici di distanza fra serie temporali. I cluster risultano di bassa qualità. Non richiede come parametro iniziale il numero di cluster. Gestisce serie temporali di lunghezze differenti attraverso l'utilizzo di misure come il DTW ma non di grandi dimensioni per via della complessità quadratica.
- **Clustering Partizionale**: necessita come parametro iniziale il numero di cluster. Gli algoritmi su cui si basa sono molto preformanti rispetto al clustering gerarchico. Sono più compatibili con misure di prossimità rispetto al tempo e con serie aventi la stessa lunghezza. Esempi: **K-means**, **K-medoids**.

Algoritmi di Clustering per Serie Temporali

Algoritmi di Clustering per Serie Temporali

- **Clustering Basato su Modelli**: ipotizza un modello per ciascuno dei cluster e trova la migliore disposizione dei dati rispetto al determinato modello. Consente di determinare automaticamente il numero di cluster. Necessita di alcuni paramatri iniziali basati su assunzioni dell'utente e risulta lento con set di dati molto grandi.

Algoritmi di Clustering per Serie Temporali

- **Clustering Basato su Modelli**: ipotizza un modello per ciascuno dei cluster e trova la migliore disposizione dei dati rispetto al determinato modello. Consente di determinare automaticamente il numero di cluster. Necessita di alcuni paramatri iniziali basati su assunzioni dell'utente e risulta lento con set di dati molto grandi.
- **Clustering Basato sulla Densità**: i cluster sono sottospazi densi di oggetti che sono separati dai sottospazi in cui gli oggetti hanno bassa densità. Non è molto utilizzato con serie temporali per la sua elevata complessità.
  Esempio: **DBSCAN**

Caratteristiche dell'Algoritmo Time Series K-means

Caratteristiche dell'Algoritmo Time Series K-means

- Tecnica di **subsequence clustering**, **partizionale** e basata su **prototipi**. Scopre iterativamente i sottospazi più rilevanti dell'intera sequenza su cui sono definite le serie temporali e successivamente esegue il clustering basandosi su quest'ultimi.

Caratteristiche dell'Algoritmo Time Series K-means

- Tecnica di **subsequence clustering**, **partizionale** e basata su **prototipi**. Scopre iterativamente i sottospazi più rilevanti dell'intera sequenza su cui sono definite le serie temporali e successivamente esegue il clustering basandosi su quest'ultimi.
- Utilizza l'approccio del **K-means**. Si calcola le distanze fra le serie temporali e i centroidi dei cluster per poi aggiornarli di conseguenza fino al raggiungimento della convergenza.

Caratteristiche dell'Algoritmo Time Series K-means

- Tecnica di **subsequence clustering**, **partizionale** e basata su **prototipi**. Scopre iterativamente i sottospazi più rilevanti dell'intera sequenza su cui sono definite le serie temporali e successivamente esegue il clustering basandosi su quest'ultimi.
- Utilizza l'approccio del **K-means**. Si calcola le distanze fra le serie temporali e i centroidi dei cluster per poi aggiornarli di conseguenza fino al raggiungimento della convergenza.
- Assegna un peso specifico ad i vari istanti di tempo (**time stamps**) che caratterizzano le serie temporali in esame, ovvero un valore che specifica l'importanza di un determinato istante di tempo per il processo di clustering.

Caratteristiche dell'Algoritmo Time Series K-means

Caratteristiche dell'Algoritmo Time Series K-means

- Assegna pesi simili ad istanti di tempo adiacenti (**smooth weights**) per rendere più significativo, durante il processo di clustering, l'ordine cronologico dei dati presenti nelle serie e di conseguenza rendere i sottospazi scoperti molto più rilevanti per il clustering di serie temporali.

Caratteristiche dell'Algoritmo Time Series K-means

- Assegna pesi simili ad istanti di tempo adiacenti (**smooth weights**) per rendere più significativo, durante il processo di clustering, l'ordine cronologico dei dati presenti nelle serie e di conseguenza rendere i sottospazi scoperti molto più rilevanti per il clustering di serie temporali.
- Offre ottimi risultati su set di serie temporali che presentano una forte correlazione in specifiche sotto sequenze.

La Funzione Obiettivo

La Funzione Obiettivo

- Sia $X = \{X_1, X_2, \ldots, X_n\}$ un set di n serie temporali. Ogni $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ è caratterizzato da m valori corrispondenti ad m istanti di tempo.

La Funzione Obiettivo

- Sia $X = \{X_1, X_2, \ldots, X_n\}$ un set di n serie temporali. Ogni $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ è caratterizzato da m valori corrispondenti ad m istanti di tempo.
- Sia U una matrice binaria di dimensioni $n \times k$, dove l'elemento $u_{ip} = 1$ indica che la serie temporale i è stata assegnata al cluster p, altrimenti $u_{ip} = 0$. k è il numero di cluster che vogliamo trovare.

La Funzione Obiettivo

- Sia $X = \{X_1, X_2, \dots, X_n\}$ un set di n serie temporali. Ogni $X_i = \{x_{i1}, x_{i2}, \dots, x_{im}\}$ è caratterizzato da m valori corrispondenti ad m istanti di tempo.
- Sia U una matrice binaria di dimensioni $n \times k$, dove l'elemento $u_{ip} = 1$ indica che la serie temporale i è stata assegnata al cluster p, altrimenti $u_{ip} = 0$. k è il numero di cluster che vogliamo trovare.
- Sia $Z = \{Z_1, Z_2, \dots, Z_k\}$ un set di k vettori, dove ogni Z_i costituisce la serie temporale definita come $Z_i = \{Z_{i1}, x_{i2}, \dots, x_{im}\}$, che descrive il centroide del cluster i.

La Funzione Obiettivo

- Sia $X = \{X_1, X_2, \ldots, X_n\}$ un set di n serie temporali. Ogni $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ è caratterizzato da m valori corrispondenti ad m istanti di tempo.
- Sia U una matrice binaria di dimensioni $n \times k$, dove l'elemento $u_{ip} = 1$ indica che la serie temporale i è stata assegnata al cluster p, altrimenti $u_{ip} = 0$. k è il numero di cluster che vogliamo trovare.
- Sia $Z = \{Z_1, Z_2, \ldots, Z_k\}$ un set di k vettori, dove ogni Z_i costituisce la serie temporale definita come $Z_i = \{Z_{i1}, x_{i2}, \ldots, x_{im}\}$, che descrive il centroide del cluster i.
- Sia $W = \{W_1, W_2, \ldots, W_k\}$ un set di k vettori che rappresenta i pesi degli istanti di tempo per ogni cluster. Il valore dell'elemento w_{pj} indica il peso associato al j-esimo istante di tempo per il p-esimo cluster.

La Funzione Obiettivo

Con le strutture precedentemente definite la funzione obiettivo che l'algoritmo TSkmeans tenta di minimizzare è formulata come segue

$$\left.P(U,Z,W) = \sum_{p=1}^{k}\sum_{i=1}^{n}\sum_{j=1}^{m} u_{ip}w_{pj}(x_{ij}-z_{pj})^2 + \quad \right\} \textit{Prima Parte}$$

$$\left.+\frac{1}{2}\alpha\sum_{p=1}^{k}\sum_{j=1}^{m-1}(w_{pj}-w_{pj+1})^2 \quad \right\} \textit{Seconda Parte}$$

soggetta a: $\begin{cases} \sum_{p=1}^{k} u_{ip} = 1, & u_{ip} \in \{0,1\} \\ \sum_{j=1}^{m} w_{pj} = 1, & 0 \leq w_{pj} \leq 1 \end{cases}$

La Funzione Obiettivo

La Funzione Obiettivo

- Il parametro α viene passato in input all'algoritmo. È utilizzato per bilanciare gli effetti della funzione obiettivo tra la dispersione delle serie temporali all'interno dei cluster e l'uniformità dei pesi associati a istanti di tempo adiacenti.

La Funzione Obiettivo

- Il parametro α viene passato in input all'algoritmo. È utilizzato per bilanciare gli effetti della funzione obiettivo tra la dispersione delle serie temporali all'interno dei cluster e l'uniformità dei pesi associati a istanti di tempo adiacenti.
- La prima parte punta a minimizzare la somma pesata della dispersione di tutti i cluster, tentando di generare dei cluster coesi e compatti. La misura di prossimità usata è una distanza Euclidea quadratica e pesata.

La Funzione Obiettivo

- Il parametro α viene passato in input all'algoritmo. È utilizzato per bilanciare gli effetti della funzione obiettivo tra la dispersione delle serie temporali all'interno dei cluster e l'uniformità dei pesi associati a istanti di tempo adiacenti.
- La prima parte punta a minimizzare la somma pesata della dispersione di tutti i cluster, tentando di generare dei cluster coesi e compatti. La misura di prossimità usata è una distanza Euclidea quadratica e pesata.
- La seconda parte punta a rendere simili i pesi per istanti di tempo adiacenti, per cercare di valorizzare maggiormente il carattere cronologico delle serie temporali e per permettere la corretta ricerca di sotto sequenze omogenee.

L'Algoritmo Time Series K-means

Algorithm 2: Time Series K-means (TSkmeans)

Input : $X = \{X_1, X_2, \ldots, X_n\}$, k, α.

Output: U, Z, W.

Inizializzazione: Sceglie in modo casuale i centroidi $Z^0 = \{Z_1, Z_2, \ldots, Z_k\}$ e i pesi $W^0 = \{W_1, W_2, \ldots, W_k\}$ iniziali.

repeat

  Fissato Z, W, ricava la matrice di appartenenza U.

  Fissato U, W, ricava la matrice dei centroidi Z.

  Fissato U, Z, ricava la matrice dei pesi W.

until *Le assegnazioni ai cluster non cambiano.*

L'Algoritmo Time Series K-means

L'Algoritmo Time Series K-means

- La fase di inizializzazione genera la matrice dei centroidi iniziali Z scegliendo casualmente k serie temporali del set. Genera in modo casuale la matrice dei pesi W rispettando le condizioni imposte.

L'Algoritmo Time Series K-means

- La fase di inizializzazione genera la matrice dei centroidi iniziali Z scegliendo casualmente k serie temporali del set. Genera in modo casuale la matrice dei pesi W rispettando le condizioni imposte.
- Nel passo di aggiornamento per ricavare la nuova matrice di appartenenza U possiamo dimostrare che fissando le matrici W e Z la funzione obiettivo è minimizzata solo se:

$$u_{ip} = \begin{cases} 1, & se\ D_{pj} \leq D_{p'j},\ p' \neq p,\ 1 \leq p' \leq k \\ 0, & altrimenti \end{cases}$$

dove $D_{pj} = \sum_{j=1}^{m} w_{pj}(x_{ij} - z_{pj})^2$.

L'Algoritmo Time Series K-means

L'Algoritmo Time Series K-means

- Nel passo di aggiornamento per ricavare la nuova matrice dei centroidi Z possiamo dimostrare che fissando le matrici U e W la funzione obiettivo è minimizzata solo se:

$$z_{pj} = \frac{\sum_{i=1}^{n} u_{ip} x_{ij}}{\sum_{i=1}^{n} u_{ip}}.$$

L'Algoritmo Time Series K-means

- Nel passo di aggiornamento per ricavare la nuova matrice dei centroidi Z possiamo dimostrare che fissando le matrici U e W la funzione obiettivo è minimizzata solo se:

$$z_{pj} = \frac{\sum_{i=1}^{n} u_{ip} x_{ij}}{\sum_{i=1}^{n} u_{ip}}.$$

- Per ricavare la nuova matrice dei pesi W fissiamo le matrici U e Z e minimizziamo la funzione obiettivo risolvendo un problema di programmazione quadratica, un problema di ottimizzazione matematica di una funzione quadratica basata su diverse variabili soggette a vincoli lineari. Utilizziamo dunque un risolutore per problemi di programmazione quadratica (**quadprog** in Matlab).

Funionalità dei Pesi

Data la relativa uniformità delle serie temporali, il TSkmeans cerca di estrarre dei sottospazi altrettanto uniformi, cercando di assegnare dei pesi simili a istanti di tempo adiacenti. Inoltre:

Funionalità dei Pesi

Data la relativa uniformità delle serie temporali, il TSkmeans cerca di estrarre dei sottospazi altrettanto uniformi, cercando di assegnare dei pesi simili a istanti di tempo adiacenti. Inoltre:

- Identificano gli istanti di tempo che hanno un alto valore discriminante per migliorare le prestazioni e i risultati del clustering.

Funionalità dei Pesi

Data la relativa uniformità delle serie temporali, il TSkmeans cerca di estrarre dei sottospazi altrettanto uniformi, cercando di assegnare dei pesi simili a istanti di tempo adiacenti. Inoltre:

- Identificano gli istanti di tempo che hanno un alto valore discriminante per migliorare le prestazioni e i risultati del clustering.
- Identificano gli intervalli di tempo dove le serie temporali presentano pattern simili facilitando dunque la loro analisi.

Funionalità dei Pesi

Data la relativa uniformità delle serie temporali, il TSkmeans cerca di estrarre dei sottospazi altrettanto uniformi, cercando di assegnare dei pesi simili a istanti di tempo adiacenti. Inoltre:

- Identificano gli istanti di tempo che hanno un alto valore discriminante per migliorare le prestazioni e i risultati del clustering.
- Identificano gli intervalli di tempo dove le serie temporali presentano pattern simili facilitando dunque la loro analisi.

L'algoritmo TSkmeans cerca di assegnare pesi più grandi ad istanti di tempo adiacenti che presentano una minore dispersione all'interno del cluster, mentre assegna pesi più piccoli a quelli che presentano una dispersione del cluster più alta.

Il Parametro α

Il parametro α regola l'uniformità fra pesi di istanti di tempo adiacenti.

Il Parametro α

Il parametro α regola l'uniformità fra pesi di istanti di tempo adiacenti.

- Se $\alpha = 0$, la seconda parte della funzione obiettivo si annulla, non otteniamo dei pesi uniformi per istanti di tempo adiacenti e all'istante che presenta la minima dispersione all'interno del cluster verrà associato un peso pari a 1, 0 a tutti gli altri.

Il Parametro α

Il parametro α regola l'uniformità fra pesi di istanti di tempo adiacenti.

- Se $\alpha = 0$, la seconda parte della funzione obiettivo si annulla, non otteniamo dei pesi uniformi per istanti di tempo adiacenti e all'istante che presenta la minima dispersione all'interno del cluster verrà associato un peso pari a 1, 0 a tutti gli altri.
- Se $\alpha < 0$, la seconda parte della funzione obiettivo risulterà negativa, portando i pesi relativi ad istanti di tempo adiacenti ad oscillare in modo marcato.

Il Parametro α

Il parametro α regola l'uniformità fra pesi di istanti di tempo adiacenti.

- Se $\alpha = 0$, la seconda parte della funzione obiettivo si annulla, non otteniamo dei pesi uniformi per istanti di tempo adiacenti e all'istante che presenta la minima dispersione all'interno del cluster verrà associato un peso pari a 1, 0 a tutti gli altri.
- Se $\alpha < 0$, la seconda parte della funzione obiettivo risulterà negativa, portando i pesi relativi ad istanti di tempo adiacenti ad oscillare in modo marcato.
- Se $\alpha > 0$, il valore della seconda parte della funzione obiettivo aumenterà con l'aumentare del valore di α, ottenendo istanti di tempo adiacenti con pesi omogenei e un clustering ottimale rispetto agli obiettivi da raggiungere.

Complessità del Time Series K-means

L'algoritmo TSkmeans è un metodo iterativo che basa la sua esecuzione su tre passaggi fondamentali:

Complessità del Time Series K-means

L'algoritmo TSkmeans è un metodo iterativo che basa la sua esecuzione su tre passaggi fondamentali:

- Aggiornamento della matrice di appartenenza U.

$$\text{Costo} = O(k * n * m).$$

Complessità del Time Series K-means

L'algoritmo TSkmeans è un metodo iterativo che basa la sua esecuzione su tre passaggi fondamentali:

- Aggiornamento della matrice di appartenenza U. Costo $= O(k * n * m)$.
- Aggiornamento della matrice dei centroidi Z. Costo $= O(k * n * m)$.

Complessità del Time Series K-means

L'algoritmo TSkmeans è un metodo iterativo che basa la sua esecuzione su tre passaggi fondamentali:

- Aggiornamento della matrice di appartenenza U. Costo $= O(k * n * m)$.
- Aggiornamento della matrice dei centroidi Z. Costo $= O(k * n * m)$.
- Aggiornamento della matrice dei pesi W. Costo $=$ Costo del risolutore $= f(j)$.

Complessità del Time Series K-means

L'algoritmo TSkmeans è un metodo iterativo che basa la sua esecuzione su tre passaggi fondamentali:

- Aggiornamento della matrice di appartenenza U. — Costo $= O(k * n * m)$.
- Aggiornamento della matrice dei centroidi Z. — Costo $= O(k * n * m)$.
- Aggiornamento della matrice dei pesi W. — Costo $=$ Costo del risolutore $= f(j)$.

La complessità totale è pari a $O(I * (f(j) + k * n * m))$ dove I rappresenta il numero di iterazioni impiegate dall'algoritmo TSkmeans per raggiungere la convergenza.

Valutazione delle Tecniche di Clustering

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

- Determinare la tendenza del cluster, ovvero scoprire se esistono delle reali strutture non casuali nei dati.

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

- Determinare la tendenza del cluster, ovvero scoprire se esistono delle reali strutture non casuali nei dati.
- Determinare il numero corretto di cluster presenti.

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

- Determinare la tendenza del cluster, ovvero scoprire se esistono delle reali strutture non casuali nei dati.
- Determinare il numero corretto di cluster presenti.
- Valutare l'adeguatezza dei risultati del clustering rispetto ai dati senza l'ausilio di informazioni esterne.

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

- Determinare la tendenza del cluster, ovvero scoprire se esistono delle reali strutture non casuali nei dati.
- Determinare il numero corretto di cluster presenti.
- Valutare l'adeguatezza dei risultati del clustering rispetto ai dati senza l'ausilio di informazioni esterne.
- Confrontare i risultati del clsutering con informazioni esterne conosciute (etichette di classe).

Valutazione delle Tecniche di Clustering

- Risulta molto difficile trovare misure indipendenti e affidabili.
- Non esiste uno metodo unico, ma una grande varietà di misure e indici presi spesso da altre aree di studio.

Alcuni aspetti da considerare per la valutazione:

- Determinare la tendenza del cluster, ovvero scoprire se esistono delle reali strutture non casuali nei dati.
- Determinare il numero corretto di cluster presenti.
- Valutare l'adeguatezza dei risultati del clustering rispetto ai dati senza l'ausilio di informazioni esterne.
- Confrontare i risultati del clsutering con informazioni esterne conosciute (etichette di classe).
- Confrontare due set di cluster per determinare qual è il migliore.

Classificazione delle Misure di Valutazione

Le misure di valutazione, chiamate anche indici, sono generalmente classificate in due tipologie.

Classificazione delle Misure di Valutazione

Le misure di valutazione, chiamate anche indici, sono generalmente classificate in due tipologie.

- **Misure Non Supervisionate** (**Indici Interni**): valutano la qualità del clustering senza riferirsi ad informazioni esterne.
  - **Misure di Coesione**: determinano quanto gli oggetti contenuti nei cluster sono correlati fra loro.
  - **Misure di Separazione**: determinano quanto un cluster sia distinto o separato da altri cluster.

Classificazione delle Misure di Valutazione

Le misure di valutazione, chiamate anche indici, sono generalmente classificate in due tipologie.

- **Misure Non Supervisionate** (**Indici Interni**): valutano la qualità del clustering senza riferirsi ad informazioni esterne.
  - **Misure di Coesione**: determinano quanto gli oggetti contenuti nei cluster sono correlati fra loro.
  - **Misure di Separazione**: determinano quanto un cluster sia distinto o separato da altri cluster.
- **Misure Supervisionate** (**Indici Esterni**): valutano quanto corrisponde un pattern trovato da un algoritmo di clustering rispetto ad una struttura fornita esternamente. Utilizzano delle informazioni che non sono presenti nel set di dati, come le etichette di classe.

Coesione e Separazione

Coesione e Separazione

Per cluster basati su prototipi la coesione di un cluster è definita come la somma delle prossimità rispetto al prototipo del cluster.

Coesione(C_i): $\sum_{x \in C_i} \text{prossimità}(x, c_i)$
dove c_i rappresenta il prototipo del cluster C_i

Coesione e Separazione

Per cluster basati su prototipi la coesione di un cluster è definita come la somma delle prossimità rispetto al prototipo del cluster.

Coesione(C_i): $\sum_{x \in C_i} \text{prossimità}(x, c_i)$
dove c_i rappresenta il prototipo del cluster C_i

La separazione di un cluster invece è correlata alla separazione tra il prototipo dei cluster e un prototipo complessivo $\mathcal{C}$ calcolato rispetto ad ogni oggetto del set di dati, ovvero:

Separazione(C_i): $\text{prossimità}(c_i, \mathcal{C})$

Indici Interni

Indici Interni

Misura di Coesione

Se scegliamo come prossimità la distanza euclidea quadratica, otteniamo l'indice interno **SSE**: $\sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.

Indici Interni

Misura di Coesione

Se scegliamo come prossimità la distanza euclidea quadratica, otteniamo l'indice interno **SSE**: $\sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.

Misura di Separazione

Se scegliamo come prossimità la distanza euclidea standard, otteniamo l'indice interno **SSB**: $\sum_{i=1}^{K} m_i * dist(c_i, \mathcal{C})^2$.

Indici Interni

Misura di Coesione
Se scegliamo come prossimità la distanza euclidea quadratica, otteniamo l'indice interno **SSE**: $\sum_{i=1}^{K} \sum_{X \in \mathcal{C}_i} dist(\mathbb{C}_i, X)^2$.
Misura di Separazione
Se scegliamo come prossimità la distanza euclidea standard, otteniamo l'indice interno **SSB**: $\sum_{i=1}^{K} m_i * dist(c_i, \mathcal{C})^2$.

Indici Interni Globali (combinazione tra coesione e separazione)

- $\mathbf{TSS} = \mathbf{SSE} + \mathbf{SSB}$.
- **Coefficiente di Silhouette**: $\frac{b_i - a_i}{\max(a_i, b_i)}$, con a_i pari alla distanza media dell'oggetto da tutti gli altri appartenenti al suo cluster e b_i pari alla minima distanza tra le medie delle distanze tra esso e gli oggetti contenuti in cluster che non lo contengono.

Indici Esterni

Gli indici esterni valutano il clustering misurando il grado di corrispondenza tra le etichette dei cluster finali restituiti da un algoritmo di clustering e le etichette di classe fornite esternamente.

Indici Esterni

Gli indici esterni valutano il clustering misurando il grado di corrispondenza tra le etichette dei cluster finali restituiti da un algoritmo di clustering e le etichette di classe fornite esternamente.

Gli indici che presenteremo seguono due approcci:

Indici Esterni

Gli indici esterni valutano il clustering misurando il grado di corrispondenza tra le etichette dei cluster finali restituiti da un algoritmo di clustering e le etichette di classe fornite esternamente.

Gli indici che presenteremo seguono due approcci:

- **Orientati alla Classificazione**: valutano in quale misura un cluster contiene oggetti di una singola classe.

Indici Esterni

Gli indici esterni valutano il clustering misurando il grado di corrispondenza tra le etichette dei cluster finali restituiti da un algoritmo di clustering e le etichette di classe fornite esternamente.

Gli indici che presenteremo seguono due approcci:

- **Orientati alla Classificazione**: valutano in quale misura un cluster contiene oggetti di una singola classe.
- **Orientati alla Similarità**: valutano in quale misura due oggetti che appartengono alla stessa classe si trovano nello stesso cluster e vice versa.

Indici Esterni

Supponiamo che $C = \{C_1, C_2, \ldots, C_K\}$ sia il set dei cluster restituiti dall'algoritmo di clustering che vogliamo valutare e $C' = \{C'_1, C'_2, \ldots, C'_K\}$ sia il set delle classi dei dati, ovvero, il set ricavato dalle etichette di classe.

Indici Esterni

Supponiamo che $C = \{C_1, C_2, \ldots, C_K\}$ sia il set dei cluster restituiti dall'algoritmo di clustering che vogliamo valutare e $C' = \{C'_1, C'_2, \ldots, C'_K\}$ sia il set delle classi dei dati, ovvero, il set ricavato dalle etichette di classe.

- **Purity**: $\frac{1}{N} \sum_{i=1}^{K} \max_{1 \leq j \leq K} |C_i \cap C'_j|$.

Indici Esterni

Supponiamo che $C = \{C_1, C_2, \dots, C_K\}$ sia il set dei cluster restituiti dall'algoritmo di clustering che vogliamo valutare e $C' = \{C'_1, C'_2, \dots, C'_K\}$ sia il set delle classi dei dati, ovvero, il set ricavato dalle etichette di classe.

- **Purity**: $\frac{1}{N}\sum_{i=1}^{K} \max_{1\leq j\leq K} |C_i \cap C'_j|$.
- **F-score**: $\sum_{j=1}^{k} \frac{n_j}{N} \max_{1\leq i\leq k} \frac{2*(n_{i,j}/n_j)*(n_{i,j}/n_i)}{n_{i,j}/n_j+n_{i,j}/n_i}$.

Indici Esterni

Supponiamo che $C = \{C_1, C_2, \ldots, C_K\}$ sia il set dei cluster restituiti dall'algoritmo di clustering che vogliamo valutare e $C' = \{C'_1, C'_2, \ldots, C'_K\}$ sia il set delle classi dei dati, ovvero, il set ricavato dalle etichette di classe.

- **Purity**: $\frac{1}{N}\sum_{i=1}^{K} \max_{1\leq j\leq K} |C_i \cap C'_j|$.
- **F-score**: $\sum_{j=1}^{k} \frac{n_j}{N} \max_{1\leq i\leq k} \frac{2*(n_{i,j}/n_j)*(n_{i,j}/n_i)}{n_{i,j}/n_j+n_{i,j}/n_i}$.
- **Rand Index**: $\frac{a+d}{M}$.

Indici Esterni

Supponiamo che $C = \{C_1, C_2, \ldots, C_K\}$ sia il set dei cluster restituiti dall'algoritmo di clustering che vogliamo valutare e $C' = \{C'_1, C'_2, \ldots, C'_K\}$ sia il set delle classi dei dati, ovvero, il set ricavato dalle etichette di classe.

- **Purity**: $\frac{1}{N}\sum_{i=1}^{K} \max_{1 \leq j \leq K} |C_i \cap C'_j|$.
- **F-score**: $\sum_{j=1}^{k} \frac{n_j}{N} \max_{1 \leq i \leq k} \frac{2*(n_{i,j}/n_j)*(n_{i,j}/n_i)}{n_{i,j}/n_j + n_{i,j}/n_i}$.
- **Rand Index**: $\frac{a+d}{M}$.
- **Normalized Mutual Information (NMI)**: $\frac{\sum_{i=1}^{K}\sum_{j=1}^{K} n_{i,j}\log_2(\frac{n*n_{i,j}}{n_i*n_j})}{\sqrt{(\sum_{i=1}^{K} n_i \log_2 \frac{n_i}{n})(\sum_{j=1}^{K} \log_2 \frac{n_j}{n})}}$.

Test su Set di Dati Sintetico

Valuteremo il TSkmeans con un set di serie temporali pensato per sfruttarne i punti di forza, ovvero un set di serie temporali che presentano una forte correlazione in alcune loro sotto sequenze.

Per fare questo:

Test su Set di Dati Sintetico

Valuteremo il TSkmeans con un set di serie temporali pensato per sfruttarne i punti di forza, ovvero un set di serie temporali che presentano una forte correlazione in alcune loro sotto sequenze.

Per fare questo:

- Generiamo un set di serie temporali sintetico creato ad hoc con le caratteristiche richieste.

Test su Set di Dati Sintetico

Valuteremo il TSkmeans con un set di serie temporali pensato per sfruttarne i punti di forza, ovvero un set di serie temporali che presentano una forte correlazione in alcune loro sotto sequenze.

Per fare questo:

- Generiamo un set di serie temporali sintetico creato ad hoc con le caratteristiche richieste.
- Ricaviamo una procedura per ottenere il valore ottimale per il parametro α, necessario al TSkmeans.

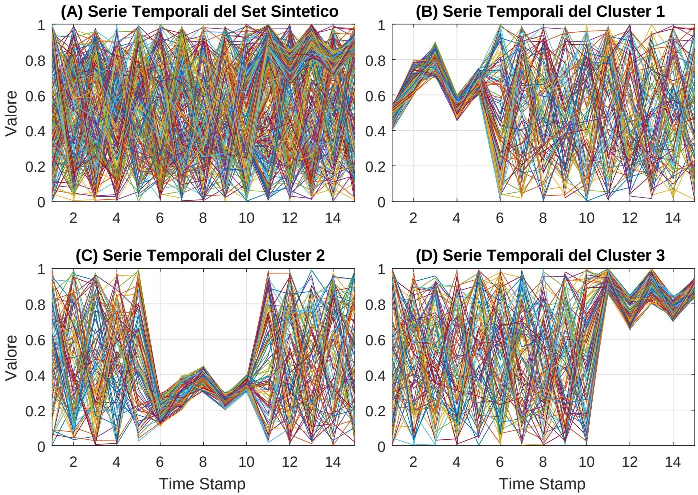
Test su Set di Dati Sintetico

Valuteremo il TSkmeans con un set di serie temporali pensato per sfruttarne i punti di forza, ovvero un set di serie temporali che presentano una forte correlazione in alcune loro sotto sequenze.

Per fare questo:

- Generiamo un set di serie temporali sintetico creato ad hoc con le caratteristiche richieste.
- Ricaviamo una procedura per ottenere il valore ottimale per il parametro α, necessario al TSkmeans.
- Confrontiamo le sue prestazioni con alcuni noti algoritmi partizionali, tra cui: **K-means** e **K-medoids** basati su varie misure di prossimità incluso il DTW.

Generazione del Set di Dati Sintetico



Ricerca del Valore Ottimale di α

Ricerca del Valore Ottimale di α

- Per trovare il valore ottimale del parametro α che consenta al TSkmeans di raggiungere le prestazioni migliori studiamo l'impatto di quest'ultimo sui risultati del processo di clustering.

Ricerca del Valore Ottimale di α

- Per trovare il valore ottimale del parametro α che consenta al TSkmeans di raggiungere le prestazioni migliori studiamo l'impatto di quest'ultimo sui risultati del processo di clustering.
- Per calibrare il valore di α rispetto alle dimensioni del set di dati in esame, assegniamo un diverso valore ad α a seconda della **dispersione globale** del set.

$$\mathbf{gs} = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - z_{oj})^2, \qquad \text{dove } z_{oj} = (\sum_{i=1}^{n} x_{ij})/n$$

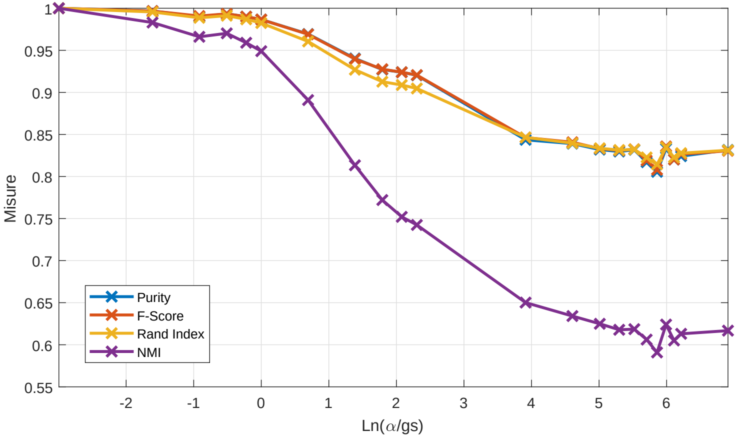
Ricerca del Valore Ottimale di α

- Per trovare il valore ottimale del parametro α che consenta al TSkmeans di raggiungere le prestazioni migliori studiamo l'impatto di quest'ultimo sui risultati del processo di clustering.
- Per calibrare il valore di α rispetto alle dimensioni del set di dati in esame, assegniamo un diverso valore ad α a seconda della **dispersione globale** del set.

$$\mathbf{gs} = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - z_{oj})^2, \qquad \text{dove } z_{oj} = (\sum_{i=1}^{n} x_{ij})/n$$

- Per i nostro obiettivi dovremo scegliere un $\alpha > 0$.

Prestazioni del TSkmeans al Variare del Parametro α



Analisi delle Prestazioni e dei Risultati sul Set Sintetico

Analisi delle Prestazioni e dei Risultati sul Set Sintetico

- Il K-means e il K-medoids producono una soluzione locale ottimale a seconda della posizione dei prototipi iniziali scelti.

Analisi delle Prestazioni e dei Risultati sul Set Sintetico

- Il K-means e il K-medoids producono una soluzione locale ottimale a seconda della posizione dei prototipi iniziali scelti.
- Eseguiremo un test basato su 100 esecuzioni di ogni algoritmo, valutando le loro prestazioni.

Analisi delle Prestazioni e dei Risultati sul Set Sintetico

- Il K-means e il K-medoids producono una soluzione locale ottimale a seconda della posizione dei prototipi iniziali scelti.
- Eseguiremo un test basato su 100 esecuzioni di ogni algoritmo, valutando le loro prestazioni.
- Durante ogni iterazione del test genereremo un set di prototipi iniziali da applicare in input ad ogni algoritmo.

Analisi delle Prestazioni e dei Risultati sul Set Sintetico

- Il K-means e il K-medoids producono una soluzione locale ottimale a seconda della posizione dei prototipi iniziali scelti.
- Eseguiremo un test basato su 100 esecuzioni di ogni algoritmo, valutando le loro prestazioni.
- Durante ogni iterazione del test genereremo un set di prototipi iniziali da applicare in input ad ogni algoritmo.
- Per le valutazioni calcoleremo la media degli indici interni ed esterni e dei tempi di esecuzione ottenuti dopo aver lanciato gli algoritmi per ogni iterazioni del test.

Analisi delle Prestazioni e dei Risultati sul Set Sintetico

- Il K-means e il K-medoids producono una soluzione locale ottimale a seconda della posizione dei prototipi iniziali scelti.
- Eseguiremo un test basato su 100 esecuzioni di ogni algoritmo, valutando le loro prestazioni.
- Durante ogni iterazione del test genereremo un set di prototipi iniziali da applicare in input ad ogni algoritmo.
- Per le valutazioni calcoleremo la media degli indici interni ed esterni e dei tempi di esecuzione ottenuti dopo aver lanciato gli algoritmi per ogni iterazioni del test.
- Per il test sul set sintetico scegliamo $\alpha = gs$ e cerchiamo tre cluster ($k = 3$).

Risultati per Indici Esterni ed Interni sul Set Sintetico

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	**0,986333333**	**0,986326628**	**0,982112821**	**0,948174643**
Euclide K-Means	0,791033333	0,793730143	0,805091193	0,575133902
Pearson K-Means	0,6607	0,641179437	0,698556745	0,365802562
Manhattan K-Means	0,7026	0,704396466	0,749279822	0,497389988
Coseno K-Means	0,692133333	0,669130401	0,717218952	0,404929828
DTW K-Means	0,876166667	0,87455308	0,850332219	0,655931307
DTW K-Medoids	0,5956	0,587754689	0,654052174	0,26367622
Euclide K-Medoids	0,566	0,584421519	0,665084504	0,333768924

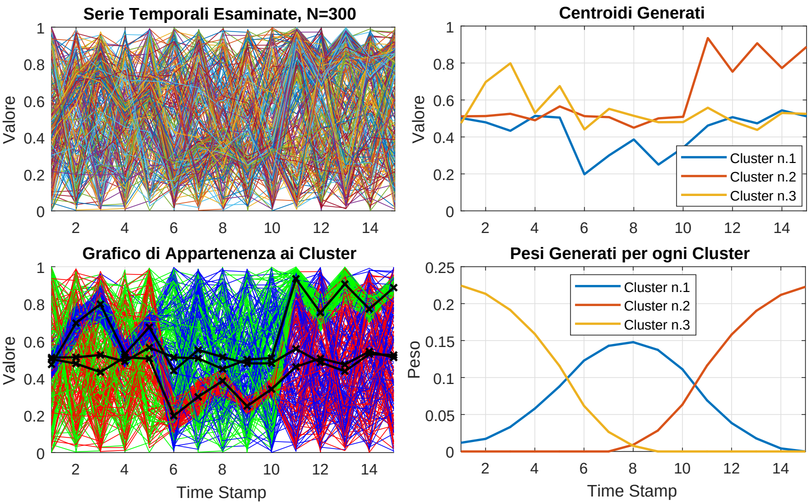
Algoritmo	SSE	SSB	TSS	Silhouette
Time Series K-Means	248,1616183	70,05222872	318,213847	0,193038274
Euclide K-Means	**245,8126212**	**72,40122581**	**318,213847**	**0,211873875**
Pearson K-Means	1545,630515	1334,612523	2880,243038	0,171888983
Manhattan K-Means	262,6800132	114,0652551	376,7452683	0,189018879
Coseno K-Means	703,3291256	433,3445472	1136,673673	0,178642311
DTW K-Means	247,995268	70,21857902	318,213847	0,196352386
DTW K-Medoids	392,2371303	184,3479062	576,5850366	0,103708787
Euclide K-Medoids	311,7709106	155,125631	466,8965415	0,159175225

Tempi di Esecuzione sul Set Sintetico

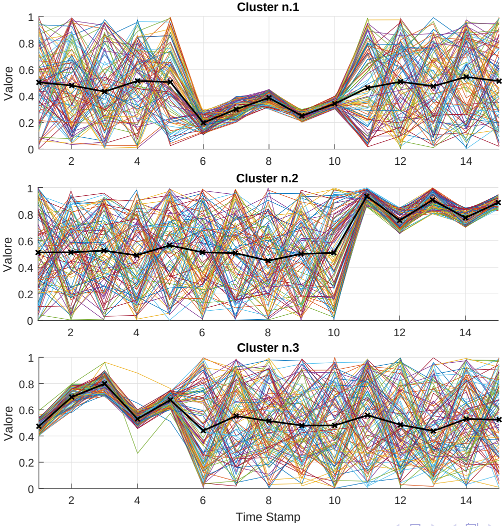
Algoritmo	Tempi di Esecuzione (secondi)
Time Series K-Means	0,018647306
Euclide K-Means	**0,002060774**
Pearson K-Means	0,00217509
Manhattan K-Means	0,002352556
Coseno K-Means	0,002056027
DTW K-Means	0,264020586
DTW K-Medoids	0,730440912
Euclide K-Medoids	0,006601104

Tempi di esecuzione medi degli algoritmi sul set sintetico

Risultati del Clustering sul Set Sintetico



Cluster Finali Trovati dal TSkmeans nel Set Sintetico



Test su Set di Dati Reali

Valuteremo il TSkmeans utilizzando cinque set di dati reali legati a problemi pratici.

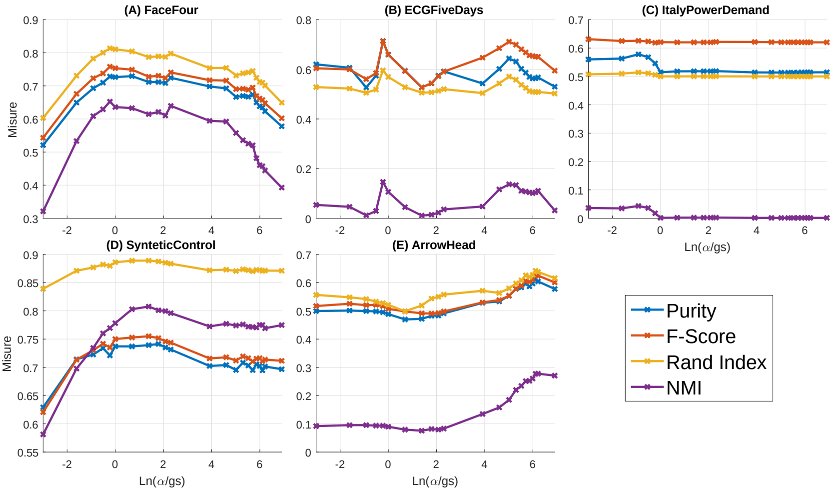
Test su Set di Dati Reali

Valuteremo il TSkmeans utilizzando cinque set di dati reali legati a problemi pratici.

Set di Dati	Serie Temporali	Istanti di Tempo	Cluster
FaceFour	350	112	4
ECGFiveDays	884	136	2
ItalianPowerDemand	1096	24	2
SynteticControl	600	60	6
ArrowHead	211	175	3

Proprietà dei set di dati reali

Ricerca del Valore Ottimale di α



Analisi delle Prestazioni e dei Risultati sui Set Reali

Analisi delle Prestazioni e dei Risultati sui Set Reali

- Per il set **FaceFour** scegliamo $\alpha = gs$ e cerchiamo quattro cluster ($k = 4$).

Analisi delle Prestazioni e dei Risultati sui Set Reali

- Per il set **FaceFour** scegliamo $\alpha = gs$ e cerchiamo quattro cluster ($k = 4$).
- Per il set **ECGFiveDays** scegliamo $\alpha = gs$ e cerchiamo due cluster ($k = 2$).

Analisi delle Prestazioni e dei Risultati sui Set Reali

- Per il set **FaceFour** scegliamo $\alpha = gs$ e cerchiamo quattro cluster ($k = 4$).
- Per il set **ECGFiveDays** scegliamo $\alpha = gs$ e cerchiamo due cluster ($k = 2$).
- Per il set **ItalianPowerDemand** scegliamo $\alpha = gs/e$ e cerchiamo due cluster ($k = 2$).

Analisi delle Prestazioni e dei Risultati sui Set Reali

- Per il set **FaceFour** scegliamo $\alpha = gs$ e cerchiamo quattro cluster ($k = 4$).
- Per il set **ECGFiveDays** scegliamo $\alpha = gs$ e cerchiamo due cluster ($k = 2$).
- Per il set **ItalianPowerDemand** scegliamo $\alpha = gs/e$ e cerchiamo due cluster ($k = 2$).
- Per il set **SynteticControl** scegliamo $\alpha = gs * e$ e cerchiamo sei cluster ($k = 6$).

Analisi delle Prestazioni e dei Risultati sui Set Reali

- Per il set **FaceFour** scegliamo $\alpha = gs$ e cerchiamo quattro cluster ($k = 4$).
- Per il set **ECGFiveDays** scegliamo $\alpha = gs$ e cerchiamo due cluster ($k = 2$).
- Per il set **ItalianPowerDemand** scegliamo $\alpha = gs/e$ e cerchiamo due cluster ($k = 2$).
- Per il set **SynteticControl** scegliamo $\alpha = gs * e$ e cerchiamo sei cluster ($k = 6$).
- Per il set **ArrowHead** scegliamo $\alpha = gs * e^6$ e cerchiamo tre cluster ($k = 3$).

Indici Esterni su FaceFour e ECGFiveDays

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	**0,711428571**	**0,740902869**	**0,800263835**	**0,62046674**
Euclide K-Means	0,619732143	0,643951615	0,73789897	0,434269448
Pearson K-Means	0,619285714	0,645743903	0,737889318	0,440086309
Manhattan K-Means	0,659375	0,683934856	0,761015122	0,494492705
Coseno K-Means	0,619285714	0,645743903	0,737889318	0,440086309
DTW K-Means	0,574196429	0,617547639	0,660217181	0,402113367
DTW K-Medoids	0,696428571	0,748823005	0,787323037	0,630183363
Euclide K-Medoids	0,623392857	0,650082077	0,732833012	0,424862644

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	**0,651414027**	**0,650444788**	**0,564724151**	**0,099148309**
Euclide K-Means	0,516606335	0,516587819	0,500001691	0,000818773
Pearson K-Means	0,515395928	0,515382646	0,499914037	0,000692331
Manhattan K-Means	0,505045249	0,505015172	0,499499905	9,53892E-05
Coseno K-Means	0,515395928	0,515382646	0,499914037	0,000692331
DTW K-Means	0,57459276	0,587454768	0,512643292	0,024396051
DTW K-Medoids	0,624174208	0,618802245	0,532767381	0,060236087
Euclide K-Medoids	0,516968326	0,516967708	0,500010249	0,000830939

Indici Esterni su ItalianPowerDemand e SynteticControl

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	0,57129562	0,624861489	0,51239556	0,040740059
Euclide K-Means	0,51459854	0,620223369	0,499970003	0,001339382
Pearson K-Means	0,51459854	0,622317201	0,499970003	0,001401493
Manhattan K-Means	**0,678439781**	**0,72152331**	**0,637524798**	**0,248742946**
Coseno K-Means	0,51459854	0,622317201	0,499970003	0,001401493
DTW K-Means	0,510072993	0,593938404	0,499746592	0,0004417
DTW K-Medoids	0,510948905	0,603176849	0,499783355	0,000579065
Euclide K-Medoids	0,51459854	0,622317201	0,499970003	0,001401493

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	0,745733333	0,761457953	0,892002393	0,811788441
Euclide K-Means	0,69685	0,710479019	0,870913634	0,774455804
Pearson K-Means	0,592183333	0,581727431	0,817860824	0,603801536
Manhattan K-Means	0,631616667	0,64583197	0,845612577	0,656017973
Coseno K-Means	0,592183333	0,581727431	0,817860824	0,603801536
DTW K-Means	0,726683333	0,751426388	0,874652031	0,728802501
DTW K-Medoids	**0,956666667**	**0,956607807**	**0,972365053**	**0,909724987**
Euclide K-Medoids	0,5517	0,546711348	0,806646689	0,58065094

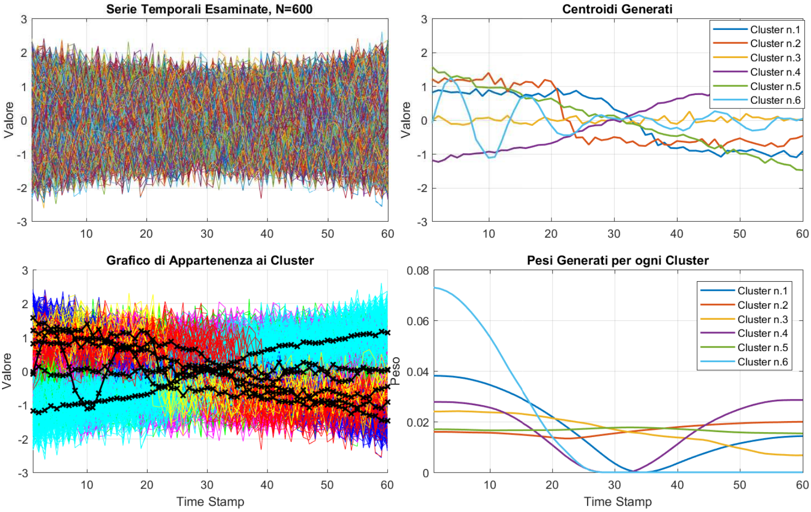
Indici Esterni su ArrowHead e Tempi di Esecuzione

Algoritmo	Purity	F-Score	Rand Index	NMI
Time Series K-Means	**0,607535545**	**0,620895227**	**0,638603024**	0,271419402
Euclide K-Means	0,565924171	0,587779848	0,607749041	0,258257701
Pearson K-Means	0,565308057	0,585057386	0,605330174	0,262193164
Manhattan K-Means	0,57943128	0,607766194	0,630229294	**0,276596756**
Coseno K-Means	0,565308057	0,585057386	0,605330174	0,262193164
DTW K-Means	0,589810427	0,599034165	0,63150079	0,24094861
DTW K-Medoids	0,588056872	0,588809678	0,627227262	0,260555797
Euclide K-Medoids	0,570236967	0,59685999	0,614284812	0,271528738

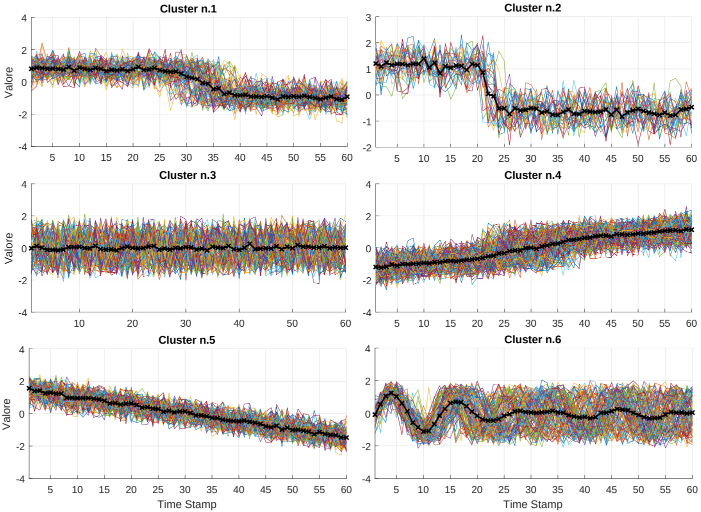
Tempi di Esecuzione (secondi)

Algoritmo	FaceFour	ECGFiveDays	ItalianPowerDemand	SynteticControl	ArrowHead
Time Series K-Means	3,657559638	0,371639306	0,05533087	0,321638485	0,576603267
Euclide K-Means	**0,002325891**	**0,005394802**	**0,00251172**	**0,002999766**	**0,003515345**
Pearson K-Means	0,002754315	0,006040031	0,002926271	0,003325851	0,003809942
Manhattan K-Means	0,006108559	0,014857353	0,004154094	0,007399809	0,00807227
Coseno K-Means	0,002631143	0,005952164	0,002884968	0,003275857	0,003706915
DTW K-Means	0,529018369	1,741742405	0,63453599	1,809497805	0,612554328
DTW K-Medoids	2,079258219	25,23698814	10,34541187	5,673538444	3,415672039
Euclide K-Medoids	0,00564299	0,053022563	0,056147643	0,047775712	0,007509423

Risultati del Clustering sul Set SynteticControl



Cluster Finali Trovati nel Set SynteticControl



Conclusioni

Conclusioni

- E' in grado di trovare sotto sequenze di serie temporali in cui si manifestano forti correlazioni fra i dati e di associare un peso maggiore a tali sequenze durante il clustering.

Conclusioni

- E' in grado di trovare sotto sequenze di serie temporali in cui si manifestano forti correlazioni fra i dati e di associare un peso maggiore a tali sequenze durante il clustering.
- Possono esserci particolari set di dati che per le loro enormi dimensioni e per via di molti dati anomali o rumori possono mettere in difficoltà l'algoritmo.

Conclusioni

- E' in grado di trovare sotto sequenze di serie temporali in cui si manifestano forti correlazioni fra i dati e di associare un peso maggiore a tali sequenze durante il clustering.
- Possono esserci particolari set di dati che per le loro enormi dimensioni e per via di molti dati anomali o rumori possono mettere in difficoltà l'algoritmo.
- Anche nei casi peggiori dei nostri test il TSkmeans riesce a mantenere delle prestazioni solo di poco inferiori rispetto agli algoritmi classici per serie temporali.

Conclusioni

- E' in grado di trovare sotto sequenze di serie temporali in cui si manifestano forti correlazioni fra i dati e di associare un peso maggiore a tali sequenze durante il clustering.
- Possono esserci particolari set di dati che per le loro enormi dimensioni e per via di molti dati anomali o rumori possono mettere in difficoltà l'algoritmo.
- Anche nei casi peggiori dei nostri test il TSkmeans riesce a mantenere delle prestazioni solo di poco inferiori rispetto agli algoritmi classici per serie temporali.

Uno spunto per un eventuale miglioramento del TSkmeans potrebbe essere la ricerca di un metodo automatico per il calcolo del valore ottimale per il parametro α.