Sommario

Questo lavoro di tesi punta a migliorare le previsioni dei valori di dati multivariati riducendo il trade-off tra accuratezza ed efficienza.

Indice

Elenco delle figure

Capitolo 1

Introduzione

La rapida evoluzione, vissuta negli ultimi decenni, per la scienza e la tecnica ci porta oggi a collezionare moli di dati impressionanti. È proprio in questi casi che soluzioni di analisi innovative che combinino informatica e statistica si rendono necessarie.

In questi termini si presenta la problematica di dover effettuare previsioni accurate di ciò che sarà l'andamento futuro di un flusso di dati multi-variati e continuo nel tempo.

Al fine di risolvere tale problematica si utilizzano tecniche di data mining capaci di gestire tali flussi e di fornirne una rappresentazione funzionale all'applicazione di modelli di regressione che sono utili ad effettuare previsioni sull'andamento futuro dei dati in osservazione.

In questo lavoro di tesi, partendo da un algoritmo che effettua previsione dopo aver applicato una tecnica di clustering incrementale e aver appreso un modello di regressione chiamato "Vector Autoregressive Model", si investiga l'utilizzo della tecnica dello smoothing esponenziale e la costruzione di aggregati dei dati osservati in tempo reale, al fine di ottenere un bilanciamento tra accuratezza e efficienza nella costruzione di un modello di previsione multi-variato in reti di sensori.

In questo capitolo vengono descritti lo scenario applicativo all'interno del quale questo lavoro di tesi si colloca e si riporta una sintetica descrizione degli obiettivi prefissati.

1.1 Scenario applicativo

L'ormai diffuso uso delle reti di sensori ha portato a dover gestire sistemi informatizzati in grado di misurare e organizzare dati multi-variati in tempo reale. Tali sistemi si avvalgono dei cosiddetti geosensori, ovvero di dispositivi in grado di eseguire rilevazioni temporali multiple di fenomeni fisici quali per esempio temperatura, umidità, velocità del vento e livello delle radiazioni solari.

Il fine ultimo di tali rilevazioni è senza dubbio quello di dover effettuare previsioni accurate di ciò che potrà essere l'andamento futuro di tali fenomeni, ma utilizzare queste informazioni non è semplice, non perché sia complicato leggerle, ma piuttosto perché è difficile interpretarle.

È in questo contesto che l'algoritmo VARForecaster trova spazio, infatti utilizza una tecnica che ci aiuta a raffinare le informazioni derivanti dai geosensori proveniente dall'insieme di tecniche e metodologie definito data mining e che si chiama analisi dei cluster. L'utilizzazione di questa tecnica ci permette di raggruppare i geosensori in base alla loro vicinanza spaziale e alla similarità delle loro osservazioni.

Per ognuno di questi gruppi distinti, si possono costruire modelli di regressione, ovvero è possibile definire un modello secondo il quale "una variabile dipendente è modellata come una funzione delle variabili indipendenti più un termine d'errore" [2].

A causa della natura delle rilevazioni, per avere delle previsioni valide e accurate è necessario che esse siano ripetute nel tempo, portando infine ad avere una serie temporale multi-variata (che considera più di un fenomeno) e georeferenziata (cioè geograficamente distribuita).

La previsione in un serie temporale multi-variata è realizzata con l'apprendimento di un modello VAR che restituisce un modello di regressione per ciascuna variabile. Tale modello di regressione predice i valori attesi per la variabile dipendente in funzione delle osservazioni storiche collezionate tanto per la variabile dipendente, cioè da predire, quanto per le variabili indipendenti, ovvero quelle rimanenti nel sistema di variabili misurate [3]. L'apprendimento di un modello VAR per sensore è computazionalmente costoso. La alternativa è l'apprendimento di un modello VAR per gruppi cluster di sensori.

In [1] si presenta un algoritmo, VARForecaster che combina modello di clustering e modello VAR. Il clustering è realizzato in maniera incrementale sulla base di approssimazioni dei valori osservati, ovvero si forma incrementalmente una gerarchia di cluster, partendo da un nodo radice inizialmente vuoto e aggiungendo istanze una alla volta aggiornando l'albero appropria-

tamente ad ogni passaggio, anche se questo comporta la ristrutturazione dell'albero [4].

In questo lavoro di tesi si presenta una variante di tale algoritmo che: (1) valuta l'uso del meccanismo di smoothing esponenziale per determinare il modello di clustering sulla base dei valori geofisici attesi nella rete all'istante t+1 piuttosto che sulla base dei valori geofisici osservati all'istante t; (2) calcola aggregati dei dati in tempo reale (media o mediana) piuttosto che ricorrere ad approssimazioni degli stessi. Lo scopo è identificare la soluzione che realizza il miglior bilanciamento tra accuratezza e efficienza.

1.2 Obiettivi e contributi

Questo lavoro di tesi considera un algoritmo incrementale di clustering spazio-temporale multi-variato e lo estende nelle seguenti direzioni:

- definizione, nell'albero binario rappresentante il clustering, di un nuovo criterio di ricalcolo della media dei nodi creati ex-novo in fase di riapprendimento (in [1] si era definito un calcolo basato sulla media del padre per ognuno di essi);
- utilizzo, per il riapprendimento dell'albero suddetto, dell'algoritmo di smoothing esponenziale, al fine di fornire un mezzo più efficace per la previsione dei valori futuri della serie di dati (in [1] tale albero era appreso solo in base ai valori ricevuti dai sensori, presenti all'interno della finestra temporale di ogni nodo).

Capitolo 2

Stato dell'arte

2.1 Smoothing

Come definito in precedenza, a causa della natura delle rilevazioni, al fine di migliorare l'accuratezza delle previsioni è indispensabile che queste siano ripetute nel tempo, e che quindi sia fornita una serie temporale di esse.

In questo lavoro di tesi si considera il caso in cui la serie temporale è multivariata, ovvero prende in considerazione più variabili, e geo-referenziata, cioè i valori sono rilevati da sensori geograficamente distribuiti.

Una serie temporale, quindi, altro non è che una sequenza di osservazioni le quali sono state ripetute nel tempo.

In una serie temporale, così come in una qualsiasi collezione di dati temporalmente distribuita, i valori assunti dalle variabili hanno una certa variazione random.

Esistono delle tecniche per ridurre gli effetti dovuti a tali variazioni, e che se utilizzate appropriatamente, rivelano con più chiarezza i pattern nascosti presenti tra i dati. Queste sono dette tecniche di "smoothing" [5].

Lo smoothing è un processo che viene usato per la previsione di valori futuri in una serie temporale e, in statistica e nel campo dell'elaborazione digitale delle immagini, per evidenziare i pattern significativi presenti all'interno dei dati, attenuando il rumore generato da artefatti ambientali, informatici o elettronici.

Nella pratica esso è un processo in cui i valori presenti in un dataset sono mediati con i loro vicini spaziali o temporali in una serie, tipo una serie temporale, o un'immagine [6].

Ci sono diversi tipi di smoothing:

- Media mobile semplice: per la previsione vengono considerate tutte le osservazioni precedenti, alle quali viene dato lo stesso peso, indipenden-

temente dall'istante temporale in cui ci troviamo. La relativa formula per il calcolo è la seguente

$$Y_{t+1} = Y_t + Y_{t-1} + Y_{t-2} + \ ... \ + Y_{t-n+1} \text{ n}$$

- Media mobile ponderata: è una variante dello smoothing a virgola mobile ed in questo caso i valori precedenti non hanno tutti lo stesso peso, bensì viene assegnato ad ognuno di essi un fattore, chiamato peso, che varierà l'influenza di tale termine all'interno della sommatoria. La formula è così definita

$$Y_{t+1} = p_1 \ Y_t + p_2 \ Y_{t-1} + p_3 \ Y_{t-2} + \ ... \ + p_n \ Y_{t-n+1}$$

  dove $p_i > 0$ e $\sum_{i=1}^{i=n} p_i = 1$

- Esponenziale.

Nella seguente sezione è descritto lo smoothing esponenziale nel dettaglio.

2.1.1 Exponential Smoothing

Lo smoothing esponenziale nacque alla fine degli anni '50 (i pionieri furono Holt nel 1957, Brown nel 1959 e Winters nel 1960) e motivò alcuni dei metodi di previsione (forecast) di maggior successo. Le previsioni ottenute con i metodi di smoothing esponenziale sono medie ponderate delle osservazioni passate, con i pesi che diminuiscono in modo esponenziale con l'andare avanti delle osservazioni.

In altri termini, le osservazioni più recenti hanno un peso associato più alto.

Questo modo di operare genera previsioni accurate in modo più veloce e per un ampio spettro di serie temporali, il che è un grande vantaggio per le applicazioni in campo industriale [7].

Essendo l'idea di base quella di "pesare" maggiormente le osservazioni più recenti rispetto a quelle più remote nel passato, i metodi di smoothing sono capaci di adeguarsi rapidamente a variazioni improvvise nel valore della serie storica a causa di eventi che modificano la regolarità del fenomeno osservato, come per es. può essere un giorno con elevata umidità oppure particolarmente ventoso.

Esistono diversi metodi di smoothing esponenziale, che tengono conto o meno dell'esistenza di componenti tendenziali e stagionali nella serie storica in esame e sono:

- Metodo di Brown;
- Metodo di Holt;
- Metodo di Winters.

Metodo di Brown

Il **metodo di Brown** è il più semplice e la sua formula è la seguente:

$$Y_t = \alpha\, x_t + (1 - \alpha)\, Y_{t-1}$$

dove α è il fattore di smoothing, con $0 \leq \alpha \leq 1$. In altre parole Y_t è la media pesata dell'osservazione corrente x_t ed il precedente termine a cui era stato applicato lo smoothing Y_{t-1}. Il fattore di smoothing è il termine, all'interno dell'equazione, che indica quanto peso dare al fattore precedente o all'osservazione corrente.

Infatti se $\alpha = 1$ la serie in uscita allo smoothing è sempre quella ricevuta in ingresso, mentre man mano che si abbassa il valore di questo fattore si ha minor adattamento ai dati correnti fino a quando, con $\alpha = 0$, si avrà in output sempre la serie di partenza.

In questo metodo il valore Y_1 è uguale al valore della prima osservazione effettuata x_1.

Questo tipo di smoothing esponenziale è molto facile da applicare, infatti basta che due osservazioni siano disponibili per poterlo utilizzare e non tiene conto ne di eventuali trend né di stagionalità all'interno della serie temporale.

Metodo di Holt

Il **metodo di Holt** migliora il modello di Brown, infatti tiene conto dei trend presenti all'interno della serie temporale (ma non della stagionalità). La formula per calcolare la previsione con il suddetto metodo è la seguente:

$$Y_{t+1} = Y_t + m_t \tag{1}$$

dove:

- $Y_t = \alpha\ x_t + (1 - \alpha)(Y_{t-1} + m_{t-1})\ \forall\ t \geq 2$ (2)

- $m_t = \beta\ (Y_t$ - $Y_{t-1})$+$(1$ - $\beta)\ m_{t-1}\ \forall\ t \geq 2$ (3)

- $Y_1 = x_1$

- $m_1 = x_2$ - x_1

L'equazione (1) è definita equazione di predizione, la (2) equazione di livello e la (3) come equazione di trend.

Y_t denota lo stimatore della serie temporale all'istante t, m_t è uno stimatore del trend della serie all'istante t, α è, come nel modello di Brown, il parametro di smoothing per la serie e può essere $0 \leq \alpha \leq 1$ e β invece è il parametro di smoothing per il trend e può anch'esso assumere valori $0 \leq \beta \leq 1$. β deve essere ottimizzato minimizzando lo scarto quadratico medio.

Metodo di Winters

Il **metodo di Winters**, infine, è il metodo di smoothing esponenziale con più alta complessità ma che si preoccupa di tenere in considerazione, oltre che dei trend presenti all'interno della serie temporale, anche di una stagionalità di periodo L nota. Esso si calcola come segue:

$$Y_{t+1} = (Y_t + m_t)\ q_{t-L+1} \tag{1}$$

dove:

- $Y_t = \alpha\ \frac{Y_t}{q_{t-L}} + (1 - \alpha)(Y_{t-1} + m_{t-1})\ \forall\ t \geq 2$ (2)
- $m_t = \beta\ (Y_t - Y_{t-1}) + (1- \beta)m_{t-1}\ \forall\ t \geq 2$ (3)
- $q_t = \gamma\ \frac{x_t}{Y_t} + (1 - \gamma)\ q_{t-L}\ \forall\ t \geq \mathrm{L} + 1$ (4)
- $q_t = \frac{x_t}{\sum_{\tau=1}^{L} \frac{x_\tau}{L}}\ \forall\ t = 1, \ldots, \mathrm{L}$ (5)
- $Y_1 = x_1$
- $m_1 = x_2 - x_1$

L'equazione (1) è definita equazione di predizione, la (2) equazione di livello, la (3) equazione di trend, mentre la (4) e la (5) sono chiamate equazioni di periodo.

Y_t denota lo stimatore della serie temporale all'istante t, m_t è uno stimatore del trend della serie all'istante t e q_t indica uno stimatore della periodicità della serie all'istante t, il quale deve essere calcolato in due modi diversi che dipendono dall'istante temporale in cui ci si trova rispetto all'intera durata del periodo.

α è il parametro di smoothing per la serie e può essere $0 \leq \alpha \leq 1$, allo stesso modo β, che è il parametro di smoothing del trend della serie temporale, deve essere $0 \leq \beta \leq 1$, invece γ è il parametro di smoothing del periodo e può essere $0 \leq \gamma \leq 1$.

γ deve essere ottimizzato minimizzando lo scarto quadratico medio.

È possibile, nel caso lo si desideri, eliminare da una serie temporale la componente di tendenza o di stagionalità. Innanzitutto bisogna determinare la tendenza con un'analisi di regressione e la componente di stagionalità tramite scomposizione della serie, successivamente è possibile calcolare la media mobile per rimuovere la tendenza e, per differenziazioni successive ($B_t(h) = y_t$ - y_{t-h}), è possibile rimuovere la tendenza [8].

2.2 Vector Autoregressive Model

Il modello VAR (Vector Autoregressive Model) è uno dei modelli più flessibili, facili da usare e di successo per l'analisi di serie temporali multivariate. E' un'estensione naturale del modello autoregressivo univariato verso il dinamismo delle serie temporali multivariate. Il modello VAR ha dimostrato di essere molto utile per descrivere il comportamento dinamico delle serie storiche economiche e finanziarie e per le previsioni. Esso fornisce previsioni spesso più accurate rispetto a quelle dei modelli univariati, infatti queste possono essere condizionate da potenziali percorsi futuri che le variabili nel modello possono prendere. Oltre alla descrizione dei dati e la previsione, il modello VAR può essere usato per l'inferenza struttura e per l'analisi della linea di condotta [13].

L'analisi del modello VAR può essere affrontata da due punti di vista:

- dal punto di vista teorico: è in questa sezione descritta la storia di tale modello e la teoricità sulla quale è basato;
- dal punto di vista implementativo: è invece qui descritta la tecnica di utilizzazione del modello VAR in base alle necessità di questo progetto di tesi.

2.2.1 Modelli teorici

I modelli VAR sono stati definiti da Christopher Sims in uno storico articolo pubblicato su Econometria nel 1980, che proponeva una critica dei modelli strutturali di equazioni simultanee, allora il principale strumento di analisi ecometrica nell'ambito della macroeconomia.

In particolare, i modelli VAR risultano nel complesso più semplici rispetto ai modelli strutturali, e la loro performance in termini di capacità previsiva di variabili macroeconomiche appare migliore.

Tutte le n variabili date in input al modello, vengono trattate simmetricamente in senso strutturale dove ognuna di essa sarà ricavata da un'equazione che spiega la sua evoluzione in base ai ritardi di se stessa e ai ritardi di tutte le altre variabili.

Le variabili che vengono date in input al modello VAR, prendono il nome di variabili endogene e vengono raccolte in un vettore Y_t di dimensione n x 1 che ha come i-esimo elemento Y_{it} e dove quest'ultimo rappresenta l'osservazione della variabile i-esima al tempo t. Ad esempio se la variabile i-esima è la velocità del vento, allora Y_{it} è il valore della velocità del vento al tempo t [9] [10].

Un modello VAR, in generale, è un sistema di equazioni simultanee che può essere descritto mediante tre modelli teorici: reduced form (o forma ridotta), structural form (o forma strutturale) e companion form.

Reduced form

Il modello VAR in forma ridotta si presenta nella forma:

$$Y_t = \mathbf{c} + \Phi(\mathbf{L})Y_{t-1} + \varepsilon_t = \mathbf{c} + \Phi_1\, T_{t-1} + \ldots + \Phi_p\, Y_{t-p} + \varepsilon_t$$

dove, per un VAR(p) $\Phi(\mathbf{L}) = \sum_{i=0}^{p-1} \Phi_i L^i$ è un polinomio matriciale di ordine p nell'operatore di ritardo $\mathbf{L}$ (ossia, l'operatore tale che $L^i\, Y_t = Y_{t-1}$), con Φ_i che è una matrice di dimensione n x n;

Y_t è il vettore delle variabili endogene e si presenta nella forma:

$$Y_t = \begin{bmatrix} Y_{1t} \\ \vdots \\ Y_{nt} \end{bmatrix}$$

e ε_t è un vettore conforme di disturbi stocastici del tipo:

$$\varepsilon_t = \begin{bmatrix} \varepsilon_{1t} \\ \vdots \\ \varepsilon_{nt} \end{bmatrix}$$

tali che $E_{\varepsilon_t} = \mathbf{0}$ (ogni termine di disturbo ha media pari a zero) e $\mathbf{E}(\varepsilon_{it}^2) = \sigma_i^2$ con $\mathbf{i = 1, \ldots, n}$ dove σ_i^2 è la varianza i-esima ed E la media. Si osservi che gli elementi del vettore ε_t non sono necessariamente non correlati, cioè che in generale $\mathbf{E}(\varepsilon_{it}\ \varepsilon_{jt}) = \sigma_{ij} \neq \mathbf{0}$ (la deviazione standard fra due elementi di ε_t può essere diversa da zero) per elementi di ε indicizzati da **i, j** con **j** $\neq$ **i**; per contro, per ipotesi nessuna delle componenti del vettore ε esibisce correlazione seriale, ossia $\mathbf{E}(\varepsilon_{it}\ \varepsilon_{i\tau}) = \mathbf{0}$, $\forall i$ **e** $\forall \tau \neq \mathbf{t}$.

Infine, resta da indicare il vettore **c**, ossia il vettore delle costanti (o intercetta) che si presenta nella seguente forma:

$$\mathbf{c} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Per capire ancora meglio la reduced form, consideriamo il modello VAR dal punto di vista della stima dei coefficienti, in modo tale da poterlo scrivere anche nel seguente modo:

$$\begin{gathered} y_{1t} = c_1 + \varphi_{11}^{(1)}\ y_{1t-1} + \ldots + \varphi_{1p}^{(p)}\ y_{nt-p} + \varepsilon_{1t} \\ \vdots \\ y_{nt} = c_n + \varphi_{n1}^{(1)} y_{1t-1} + \ldots + \varphi_{np}^{(p)} y_{nt-p} + \varepsilon_{nt} \end{gathered}$$

Osservando che al secondo membro di ogni equazione figurano le stesse variabili, il VAR(p) risulta equivalente ad un modello SURE (Seemingly Unrelated Regression Equations), i cui coefficienti possono essere stimati considerando ogni equazione come una regressione lineare standard, indipendente dalle altre. In particolare, gli stimatori OLS (Ordinary Least Squares) ottenuti con il metodo dei minimi quadrati, risultano consistenti e cioè che all'aumentare dell'informazione, la distribuzione di probabilità di ognuna di esso, si concentra in corrispondenza del valore del parametro da stimare [11].

Structural form

La structural form di un modello VAR(p) è formalizzata da un'equazione del tipo:

$$A_o\ Y_t = \mathbf{m} + A\ (\mathbf{L})\ Y_{t-1} + u_t$$

dove **m** è simile al vettore di costanti **c** della reduced form, A_0 è la matrice n x n che identifica le relazioni strutturali contemporanee tra le diverse componenti di Y_t, mentre il vettore u_t è chiamato rumore bianco ed è un vettore di disturbi che in particolare ha componenti tra loro correlate: $\mathbf{E}(u_{it}\ u_{ij}) = \mathbf{0}$ per $\mathbf{j \neq i}$.

Come si può notare, la forma strutturale è molto simile a quella ridotta con la differenza di avere la matrice A_0 al primo membro: all'interno di essa, e precisamente, sulla sua diagonale principale, si avranno elementi tutti pari a 1 e quindi vuol dire che il valore 1 lo si avrà alla variabile i-esima in corrispondenza dell'equazione i-esima.

La structural form è una forma di supporto per la reduced form, e questo perché possiamo passare dalla prima forma alla seconda pre-moltiplicando a sinistra di ambo i membri l'inversa della matrice A_0:

$$\mathbf{Y}_t{=}A_0^{-1}\ \mathbf{m}{+}A_0^{-1}\ A(L)\ \mathbf{Y}_{t-1}{+}A_0^{-1}u_t{=}\mathbf{c}{+}\Phi(L)\ \mathbf{Y}_{t-1}{+}\varepsilon$$

Quest'ultima la si può riscrivere in questo modo:

$$(I-\Phi(L)L)\ \mathbf{Y}_t{=}\mathbf{c}{+}A_0^{-1}\ u_t$$

da cui si ottiene una forma detta forma finale del modello VAR(p) o rappresentazione di Wold:

$$\mathbf{Y}_t{=}(I-\Phi(L)L)^{-1}\,\mathbf{c}+(I-\Phi(L)L)^{-1}\ A_0^{-1}\ u_t{=}\mu{+}\Psi\mathbf{(L)}u_t$$

dove $\mathbf{I}$ è la matrice identica, Ψ $\mathbf{(L)}$ è un polinomio matriciale nell'operatore L di ordine finito, e μ è il valore atteso non condizionato di Y_t. In altre parole, il VAR(p), processo vettoriale autoregressivo di ordine finito, è equivalente a un processo in media mobile di ordine infinito.

Companion form

La companion form di un modello VAR(p) è la riscrittura della sua reduced form con l'accorpamento di p espressioni vettoriali. L'equazione rappresentante questa nuova forma è la seguente:

$$\begin{bmatrix}\mathbf{Y}_t\\ \mathbf{Y}_{t-1}\\ \vdots\\ \mathbf{Y}_{t-p+1}\end{bmatrix}=\begin{bmatrix}\mathbf{c}\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}+\begin{bmatrix}\Phi_1 & \Phi_2 & \cdots & \Phi_{p-1} & \Phi_p\\ I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & I & \cdots & \mathbf{0} & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & I & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{Y}_{t-1}\\ \mathbf{Y}_{t-2}\\ \vdots\\ \mathbf{Y}_{t-p}\end{bmatrix}+\begin{bmatrix}A_0^{-1}u_t\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}$$

Seguendo la notazione prima descritta, si possono identificare, in questa equazione, quattro vettori di vettori e una matrice di matrici, dove gli zeri presenti nei vettori e nella matrice rappresentano rispettivamente vettori nulli di dimensione n x 1 e matrici nulle di dimensioni n x n.

È possibile riscrivere l'equazione nel seguente modo:

$$Z_t\ =\ \lambda\ +\ F|Z_{t-1}\ +\ \xi_t$$

dove $\mathrm{E}(\xi_t) = 0$, ed essendo $\xi_t = [\xi_1', \ldots, \xi_T']$', si ha $\mathrm{E}(\xi\ \xi') = \sum \otimes I_t$ dove $\sum$ è la matrice varianze-covarianze dei disturbi ε e $\otimes$ indica il prodotto di Kroenecker.

È importante sottolineare, infine, che la forma adottata generalmente da un modello VAR per effettuare analisi statistiche dei dati, è la **reduced form**.

2.2.2 Implementazione in R

R è un linguaggio e un ambiente di sviluppo per l'analisi statistica e grafica.

È un progetto GNU simile al linguaggio e ambiente di sviluppo S, il quale fu sviluppato ai laboratori Bell (Bell Laboratories) da John Chambers e i suoi colleghi. R può essere considerato come una diversa implementazione di S. Ci sono importanti differenze, ma molto del codice scritto in S può essere eseguito senza dover apporre modifiche in R.

R fornisce un'ampia varietà di tecniche statistiche (modellazione lineare e non, test statistici classici, analisi di serie temporali, classificazione, clustering, ...) e grafiche ed è molto estendibile.

Le sue caratteristiche principali sono:

- semplicità nella gestione e manipolazione dei dati;
- disponibilità di una suite di strumenti per calcoli su vettori, matrici, ed altre operazioni complesse;
- accesso ad un vasto insieme integrati per l'analisi statistica;
- produzione di numerose potenzialità grafiche particolarmente flessibili;
- possibilità di adoperare un vero e proprio linguaggio di programmazione orientato ad oggetti che consente l'uso di strutture condizionali e cicliche, nonché di funzioni create dall'utente;

È distribuito gratuitamente sotto i vincoli GPL (General Public License) ed è disponibile per diverse architetture hardware e sistemi operativi: Unix, Linux, Windows, MacOS. Sul sito della CRAN (`<http:\www.r-project.org) >` è possibile scaricare, oltre che il programma base, anche una serie di moduli aggiuntivi e un'ampia manualistica sull'ambiente, che va dall'installazione del software al suo utilizzo all'analisi dei dati [12].

Sono di seguito descritte le due funzioni in R considerate nella realizzazione di questo progetto di tesi:

- VARselect;
- VAR;

VARSelect

VARselect(y, lag.max = value1, type = value2, season = value3, exogen = value4)

dove:

- y è la matrice delle variabili endogene (cap. 2.1);
- lag.max, ossia il valore value1, è il massimo ordine di ritardo che si potrà avere nel modello VAR;
- type, nonché value2, rappresenta il tipo di regressori deterministici da includere nel modello VAR e uno dei possibili valori che può essere assegnato è: "const" (costante), "trend" (tendenza), "both" (entrambi, cioè costante e tendenza) e "none" (nessuno);
- season è la frequenza stagionale (periodicità) dei dati. Se value3 = 4 vorrebbe dire che si avranno dei dati trimestrali. Può assumere valore "NULL" e in questo caso indicherebbe un'assenza di stagionalità;
- exogen, ovvero value 4, rappresenta la matrice delle variabili esogene, cioè delle variabili indipendenti. Può assumere valore "NULL" e sta ad indicare che non ci sono variabili esogene.

Gli output sono:

- selection. È il vettore degli ordini ottimali ottenuti ognuno da ogni criterio possibile. I criteri sono delle funzioni avente un solo input e i criteri possibili sono quattro;
- criteria. È la matrice di dimensione 4 x lag.max contenente i valori ottenuti da ogni criterio attribuendo input che vanno da 1 fino a lag.max.

È di seguito riportato un esempio d'uso di questa funzione, con l'utilizzo del dataset "Canada" predefinito in R [20].

Questo dataset riporta le rilevazioni ottenute su quattro attributi effettuate trimestralmente dal 1980 al 2000. Tali attributi sono:

- e
- prod
- rw
- U

```
> data(Canada)
> select <- VARselect(Canada, lag.max = 5, type = "none", season = 3, exogen = NULL)
> select
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      2      2      3

$criteria
                1           2           3           4           5
AIC(n) -5.558469444 -6.15197629 -6.228779625 -6.054786702 -5.853594337
HQ(n)  -5.270082618 -5.67133158 -5.555877031 -5.189626223 -4.796175974
SC(n)  -4.838637185 -4.95225586 -4.549171021 -3.895289925 -3.214209388
FPE(n)  0.003859193  0.00214093  0.002001913  0.002424274  0.003048318
```

Figura 2.1: Esempio di utilizzo della funzione VARSelect

VAR

VAR(y, p = value1, type = value2, season = value3, exogen = value4, lag.max = value5, ic = value6)

dove i parametri già specificati per la funzione VARselect valgono anche qui mentre:

- p, nonché value1, rappresenta l'ordine di ritardo scelto per il modello VAR;
- ic, cioè value6, è l'informazione sul criterio utilizzato. I possibili criteri sono: "AIC" (Akaike), "HQ" (Hannan-Quinn), "SC" (Schwarz), "FPE" (Forecast Predict Error).

Gli output invece sono:

- p. È l'ordine di ritardo dato in input;
- K. Rappresenta la dimensione del modello VAR (cioè è il numero di attributi della matrice y data in input alla funzione VAR);
- varresult. È la lista di oggetti di regressione, dove ogni oggetto sarà associato al rispettivo attributo della matrice y data in input alla funzione VAR. Ad ognuno di questi oggetti è associato un vettore di coefficienti stimati tramite la regressione che si chiama coefficients. Questo vettore avrà un numero di coefficienti pari a p x K [13].

Come esempio di utilizzo di questa funzione si mostra la continuazione dell'apprendimento del modello VAR sul dataset Canada.

```
> p <- as.numeric(select$selection[2])
> result <- VAR(Canada, p = p, type = "none", ic = "HQ")
> result
```

Figura 2.2: Esempio di utilizzo della funzione VAR

In fig. 2.2, dalla posizione scelta all'interno del vettore in output dalla funzione *"VARselect"*, *"selection"*, per la lettura di p, è possibile notare come il criterio utilizzato sia *"HQ"*.

Inoltre è ben visibile l'utilizzo di soli quattro parametri per il corretto utilizzo della funzione VAR, e ciò è possibile perché in precedenza è stata

utilizzata la funzione *VARselect* per la determinazione degli ordini di ritardo ottimali per ogni criterio.

```
VAR Estimation Results:
=======================

Estimated coefficients for equation e:
======================================
Call:
e = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2

       e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2       rw.l2        U.l2
 1.62046761  0.17973134 -0.04425592  0.11310425 -0.64815156 -0.11683270  0.04475537 -0.06581206


Estimated coefficients for equation prod:
=========================================
Call:
prod = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2

       e.l1     prod.l1       rw.l1        U.l1        e.l2     prod.l2       rw.l2        U.l2
-0.19389053  1.16559603  0.07426648 -0.66412399  0.20141693 -0.19089450 -0.06904805  0.77427171


Estimated coefficients for equation rw:
=======================================
Call:
rw = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2

        e.l1      prod.l1        rw.l1         U.l1        e.l2      prod.l2        rw.l2         U.l2
-0.273036691 -0.078046604  0.900047886 -0.024808893  0.331264372 -0.008858991  0.062587364 -0.175795886


Estimated coefficients for equation U:
======================================
Call:
U = e.l1 + prod.l1 + rw.l1 + U.l1 + e.l2 + prod.l2 + rw.l2 + U.l2

        e.l1      prod.l1        rw.l1         U.l1        e.l2      prod.l2        rw.l2         U.l2
-0.561791776 -0.091739246 -0.001960487  0.785638638  0.574926136  0.068715871 -0.002926763  0.145852929
```

Figura 2.3: Output della funzione in fig. 2.2

In fig. 2.3 è possibile notare che il modello VAR appreso combina i valori presenti all'interno del dataset con una serie di coefficienti in modo tale da costruire una combinazione lineare necessaria per il calcolo dei valori futuri di ogni attributo. Si specifica che la struttura di ogni addendo all'interno delle equazioni è definita come:

- nome dell'attributo;
- .l;
- la quantità di istanti temporali di cui bisogna spostarsi indietro per recuperare il valore da moltiplicare con il rispettivo coefficiente.

Si noti inoltre che ogni attributo è messo in relazione con tutti gli altri (anche con se stesso) e che il modello si sposta di due elementi indietro nella finestra temporale. Tale valore è determinato dall'ordine di ritardo ottimale per il criterio scelto (che ricordiamo essere due).

2.3 Clustering

Il *clustering* è una tecnica di apprendimento non supervisionato la quale organizza un'insieme di istanze in gruppi di similarità.

Tali gruppi, che contengono al loro interno istanze simili tra loro ma dissimili con le istanze contenute negli altri gruppi, vengono definiti ***cluster***.

Le istanze all'interno di un gruppo sono chiamate *oggetti* o *data point* e l'istanza centrale del cluster è definito *centroide*.

L'obiettivo di un clustering è quindi quello di scoprire il raggruppamento dei dati attraverso l'uso di un algoritmo di clustering e di una funzione che calcola la similarità (/dissimilarità) tra essi [16].

Sono di seguito descritti le principali tecniche di clustering:

- Clustering partizionale;
- Clustering gerarchico;
- Cluster concettuale e interpolativo.

2.3.1 Clustering partizionale

Il clustering partizionale è una tecnica di clustering che punta a suddividere le istanze di un dataset in k cluster. Tale numero è a discrezione dell'utente e può essere $1 \leq \mathrm{k} \leq |n|$ con |n| cardinalità massima dell'insieme delle istanze [17].

L'algoritmo più utilizzato in letteratura che implementa questa tecnica è il k-means.

Il passo base della procedura prevede la costruzione di k cluster, ognuno contenente un solo oggetto, scelto in base ad alcuni criteri. Esso è il textitcentroide del cluster e ne è rappresentativo. Per definire a quale gruppo appartengono gli oggetti rimanenti viene calcolata la distanza fra ognuno di questi e ogni centroide. L'oggetto viene quindi raggruppato nel cluster il cui centroide minimizza la distanza dall'oggetto in questione.

2.3.2 Clustering gerarchico

Il clustering gerarchico ([18] è una tecnica di apprendimento non supervisionato che si occupa di raggruppare i dati a diversi livelli di profondità, stabilendo così una gerarchia fra i gruppi di oggetti simili. Possiamo rappresentare le relazioni fra i gruppi tramite una struttura dati ad albero chiamata dendrogramma (fig. 2.4).

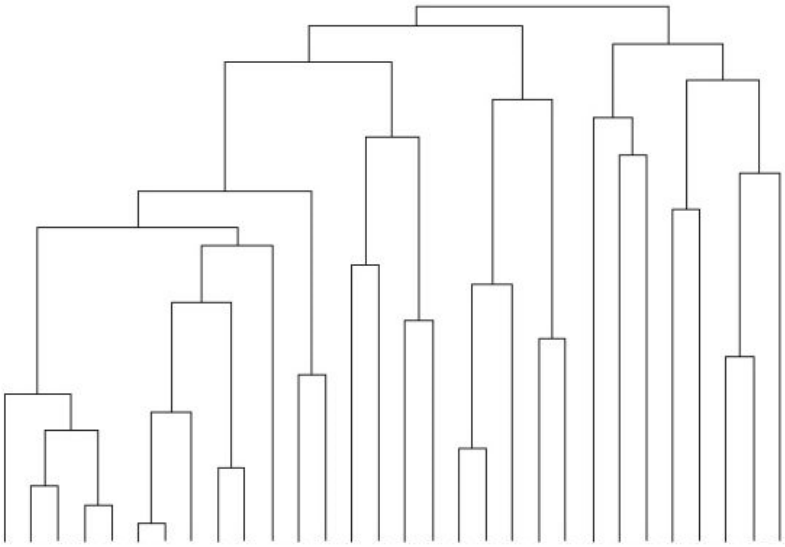

Figura 2.4: Dendrogramma

Esistono due tipi di clustering gerarchico:

- clustering divisivo: approccio di tipo top-down che parte dall'analisi dell'intero dataset, rappresentato come il nodo radice del dendrogramma e si occupa di scomporlo ricorsivamente in nodi figli, fino ad arrivare al massimo ad aggiungere nodi impossibili da partizionare ulteriormente, cioè nodi foglia costituiti ognuno da un solo oggetto.

- clustering agglomeratito: approccio di tipo bottom-up che parte dall'analisi delle foglie dell'albero, cioè le singole istanze, e costruisce un nuovo nodo, cioè il padre, degli oggetti fra loro simili entro una certa soglia. Il processo si conclude quando due o più nodi della struttura si agglomereranno in un unico nodo, che è definito *radice*, corrispondente all'intero insieme dei dati di partenza.

Il clustering gerarchico, a differenza del clustering partizionale in cui è utilizzata solo una funzione di dissimilarità tra le istanze per la costruzione di esso, necessita di una funzione di distanza al fine di calcolare quanto "distano" tra loro due insieme di oggetti, sottoinsiemi dell'insieme iniziale di dati. Questo calcolo è necessario ogni volta che si vogliono confrontare due nodi dello stesso dendogramma allo stesso livello di profondità diverso dall'ultimo. Sono di seguito elencati (e descritti concettualmente) i principali metodi per calcolare la distanza tra due insiemi:

- Metodo single-link: la distanza fra due cluster è uguale alla distanza fra i due oggetti più vicini dei due cluster (un data point per cluster). Matematicamente, questo metodo, considerando la distanza tra i cluster X e Y, è descritto tramite l'espressione:

$$D(X,Y)=\min_{x\in X,y\in Y}\ d(x,y)$$

- Metodo complete-link: la distanza fra due cluster è uguale alla distanza fra i due oggetti più lontani dei due cluster. Sempre considerando la distanza tra i cluster X e Y, la descrizione matematica di tale metodo è:

$$D(X,Y)=\max_{x\in X,y\in Y}\ d(x,y)$$

- Metodo average-link: la distanza fra due cluster è data dalla media delle distanze calcolate fra tutte le coppie di oggetti appartenenti a cluster diversi. La formula per il calcolo matematico di tale metodo è la seguente:

$$D(X,Y) = \frac{1}{|X||Y|}\ \sum_{x\in X}\ \sum_{y\in Y}\ d(x,y)$$

- Metodo dei centroidi: la distanza fra due cluster è uguale alla distanza fra i centroidi dei due cluster (X e Y). L'espressione matematica che descrive tale metodo è:

$$D(X,Y) = d(\hat{c_x}, \hat{c_y})$$

2.3.3 Clustering concettuale e interpolativo

Le tecniche tradizionali di clustering permettono di raggruppare istanze di dati che sono tra loro simili. Nel clustering concettuale o *conceptulal clustering* [19] invece si fornisce in più una interpretazione dei dati che sono stati raggruppati nello stesso insieme. Viene infatti associato a ciascun cluster la sua descrizione concettuale. In questa nuova tecnica di clustering, una nuova istanza viene confrontata con tutte le descrizioni associate ai cluster per stabilire il suo cluster di appartenenza.

Il clustering interpolativo (o *interpolative clustering*) è una forma di apprendimento supervisionato, differentemente del clustering concettuale. Oltre alla descrizione del clustering, viene associato anche un modello predittivo che sta a rappresentare un "riassunto" (*summarization*) dei dati contenuti nel cluster.

2.4 VARForecaster 1.0/2.0

In [1] sono presentate due versioni del sistema che, avuto in input un insieme di sensori (definito in seguito *dataset*), tramite applicazione del modello VAR effettuano predizione di quelli che potrebbero essere i valori assunti da tali sensori nel futuro.

2.4.1 VARForecaster 1.0

I sensori effettuano rilevazioni periodicamente e questo porta ad avere un flusso di dati multi-variato, rappresentato, per ogni istante di tempo, attraverso uno snapshot e l'insieme degli snapshot compone il dataset.

La versione VARForecaster 1.0 è caratterizzata dall'apprendimento di un modello VAR per ogni sensore appartenente alla rete, utilizzando come dati di training la serie storia multi-variata, la quale contiene le rilevazioni effettuate dal sensore.

Questo processo ([1]) avviene finché c'è uno snapshot da processare e nel dettaglio si compone di:

(1) leggere lo snapshot e memorizzarne i dati;

(2) aggiornare la serie storica multi-variata associata ad ogni sensore;

(3) apprendere un modello VAR per sensore;

(4) effettuare, se possibile, il forecast per ogni sensore.

Durante la terza fase, per ognuno dei sensori, in base ai rispettivi dati di training, viene associato un modello VAR ad ogni sensore, e ciò è possibile se e soltanto se siano stati esaminato un numero di istanti temporali sufficienti alla costruzione dello stesso. Il numero minimo di istanti per la costruzione di tale modello è parametrizzato e l'utente può deciderlo arbitrariamente. Una volta raggiunta la soglia sarà sempre possibile apprendere il modello VAR.

Se il modello VAR è stato appreso è possibile effettuare forecast.

Questo si effettua applicando, per ogni sensore, al modello VAR la serie storica dello stesso. È possibile prevedere quanti istanti temporali si vuole, con la premessa che previsioni più "lontane" sono meno precise di quelle immediatamente successive all'istante temporale t in oggetto. Per le successive previsioni le equazioni vanno calcolate considerando come ultimo istante di tempo della serie storica la previsione precedentemente effettuata. La quantità di previsioni da effettuare sono a discrezione dell'utente [1].

Le fasi di lettura del nuovo snapshot e di aggiornamento delle serie storiche devono tenere conto della probabile presenza di sensori inattivi.

Per sensore inattivo [1] si intende un sensore appartenente alla rete che non ha effettuato alcuna rilevazione in quell'istante di tempo per un qualsiasi motivo (disattivazione manuale del sensore, guasto hardware, guasto software...) e per il quale deve comunque essere mantenuta la finestra di rilevazioni effettuate nel tempo.

Durante le fasi di esecuzione del sistema tale probabilità viene gestita in maniera diversa [1]. Infatti, durante la seconda fase, ovvero quella di aggiornamento delle serie temporali viene controllato se per il sensore, precedentemente, sia stata effettuata o meno previsione. Nel caso in cui l'esito sia positivo, si sostituisce la previsione effettuata con ahead = 1 alla rilevazione (non avuta), in caso contrario si "contrassegna" tale sensore con il valore massimo che Java permette per la rappresentazione dei valori Double (definito in seguito *"MAX-VALUE"*).

In base a questa strategia è naturale che se per un sensore è effettuata una volta previsione sarà sempre possibile effettuarne altre.

Durante la seconda fase il sistema deve invece verificare che la sua serie storica non contenga il valore significativo di non rilevazione *"MAX-VALUE"*. Nel caso in cui tale valore sia presente, non è possibile effettuare apprendimento del modello VAR.

Durante la quarta ed ultima fase, il sistema deve solo controllare che per il sensore in oggetto sia stato appreso il modello VAR, in caso positivo effettuerà la previsione, altrimenti no.

2.4.2 VARForecaster 2.0

Nell'applicazione VARForecaster 2.0 ([1]) si presenta un sistema che applica il metodo del *clustering interpolativo* per ogni snapshot ricevuto in input, il quale li rappresenta mediante una struttura ad albero. Questa struttura, oltre ad includere le condizioni di split scelte in base alle autocorrelazioni spaziali che intercorrono tra i sensori, è dinamica nel tempo.

Per ciascun cluster è poi appreso un modello VAR in base alla serie storica multi-variata che lo rappresenta, determinata dalla media assunta da ciascuna variabile all'interno del cluster di appartenenza, e con la combinazione lineare tra quest'ultima ed il modello stesso è in seguito effettuata una predizione dei valori futuri per ciascuna delle istanze che ricadono all'interno di quel cluster [1].

L'autocorrelazione spaziale altro non è che un cluster territoriale di valori simili. Se a valori simili nelle istanze corrisponde una vicinanza spaziale si è in presenza di autocorrelazione spaziale positiva, al contrario l'autocorrelazione viene definita come negativa. Tali valori sono fondamentali per la scelta del partizionamento ottimale dei sensori.

I punti cardine di questa versione sono:

(1) fase di apprendimento della struttura Clustering Tree (CT in seguito);

(2) fase di modifica incrementale del CT;

(3) fase di apprendimento del modello VAR;

(4) fase di forecast.

Fasi di apprendimento e modifica incrementale del CT

Un Clustering Tree ([1]) è una struttura ad albero utilizzata per rappresentare uno snapshot mediante un insieme di cluster organizzati gerarchicamente su di esso.

In questo modo per ogni sensore ricadente all'interno del nodo in oggetto esiste una sequenza di cluster C_{i0}, C_{i1}, ..., C_{ir} nei quali la partizione all'interno del quale si trova tale sensore è inclusa. Per permettere ciò, è necessaria che sia rispettata la condizione di inclusione $C_{i0} \supseteq C_{i1} \supseteq \ldots \supseteq C_{ir}$.

Ogni cluster C_{i0}, C_{i1}, ..., C_{ir} è associato ad un nodo n_{i0}, n_{i1}, ..., n_{ir} e ogni nodo n_{ij} appartenente al CT è un figlio diretto del nodo n_{ij-1} (j = 1, *dots*, r). Il nodo radice è definito da n_{i0}, dove 0 indica appunto il livello del nodo all'interno dell'albero.

Apprendere un CT quindi vuol dire effettuare un processo di clustering sullo snapshot ricevuto in ingresso e associare una struttura ad albero ad esso, in modo tale che ogni nodo corrisponda ad un cluster.

Nel dettaglio l'algoritmo per l'apprendimento del CT lavora in maniera ricorsiva secondo un approccio top-down, ricevendo in input uno snapshot, partizionandolo (in base ai valori assunti dai sensori e alle autocorrelazioni spaziali che intercorrono tra di essi) e fermandosi quando sono soddisfatti dei criteri di stop.

Essendo questo un algoritmo ricorsivo con criterio top-down, su ogni nuovo cluster(/nodo) c trovato, si tenta di apprendere altri sotto-alberi con radice c. Tale processo si ferma se ci sono troppe poche istanze in c oppure se c'è troppo adattamento ai dati e quindi si rischia il problema dell'over-fitting dell'albero.

Il processo principale di questo algoritmo è la scelta delle condizioni di split. Essendo lo scopo fondamentale del clustering la massimizzazione della riduzione della varianza degli indicatori locali dell'autocorrelazione spaziale all'interno dei cluster, le condizioni di split indicano i migliori accoppiamenti attributo-valore in grado di diminuire tale valore. La migliore tra tutte le

condizioni è quella scelta e sarà utilizzata come decisione per dividere (partizionare) lo snapshot (o la partizione di esso se ci troviamo in un nodo che non sia la radice dell'albero).

Al termine di questa fase ad ogni nodo sarà stata associata una serie storica multi-variata, rappresentante i valori assunti nel tempo da tutti gli attributi, per ciascuna istanza che ricade nel cluster, contenente un solo valore per attributo. Tale valore è la media di tutti i valori assunti da quell'attributo da tutti i sensori presenti nel cluster.

La fase incrementale nell'apprendimento del CT lo adatta al nuovo snapshot ricevuto in input. Per fare ciò si avvale di un algoritmo ricorsivo con metodo top-down il quale attraversa l'albero e si preoccupa di trovare quei cluster che presentano un sostanziale cambiamento nella proprietà di autocorrelazione spaziale.

Questo processo di divide nella fase di pruning e nella fase di apprendimento di nuovi sotto-alberi.

La fase di pruning interessa solo i nodi di decisione e li pota se non riconosce un'effettiva convenienza nel mantenimento di tale split. Potare un nodo n significa cancellare i sotto-alberi che aveno n come radice e trasformare quindi n stesso in foglia. Tale procedimento non intacca la serie storica multi-variata associata al nodo ne tanto meno le istanze che ricadono al suo interno e quindi i valori medi, minimi e massimi assunti dalle feature nel cluster rimangono invariate.

La fase di apprendimento di nuovi sotto-alberi, successiva a quella di pruning, interessa invece soltanto i nodi foglia. Partendo da essi infatti, l'algoritmo ricorsivo cerca di capire se sia possibile e conveniente apprendere dei nuovi cluster. Nel caso in cui siano appresi nuovi cluster a partire dal nodo n, la radice del nuovo sotto-albero sarà n e le serie storiche multivariate associate a questi nodi saranno, fino all'istante *t-1* copiate dalla serie temporale multi-variata di n, mentre per l'istante temporale corrente, ovvero t, il valore sarà calcolato in base alle istanze che realmente ricadono all'interno del cluster.

Fasi di apprendimento del modello VAR e predizione

Nel momento in cui l'apprendimento incrementale di CT è terminato (per quello snapshot), per ogni cluster foglia, si prova ad apprendere un modello VAR. Questo viene appreso sulla base dei valori presenti all'interno della serie storica multi-variata ad esso associata e per poter avviare questo meccanismo vengono richiesti un minimo numero di istanti temporali all'interno di tale serie. Il numero minimo di medie multi-variate richieste all'interno della serie è arbitrario ed è stato parametrizzato. Questo perché tale decisione

dipende dalle necessità dell'utente. Tutto ciò implica che una volta raggiunta la soglia minima di istanti temporali processati è sempre possibile apprendere il modello VAR.

Nella fase di forecast vengono recuperati per ogni nodo il modello VAR ad esso associato e in base allo snapshot attuale, questo viene applicato all'istanza, quindi al sensore, corretto.

Per ciò che riguarda l'eventuale presenza di sensori inattivi, questa influisce soltanto sulla fase di previsione, in quanto, durante la fase di aggiornamento delle serie temporali ci si comporta come fatto in VARForecaster 1.0, per la fase di apprendimento dell'albero vengono considerati soltanti i sensori attivi e per l'apprendimento del modello VAR viene utilizzata la serie temporale associata al cluster e quindi eventuali valori non rilevati, non facendo parte di tale cluster, non ne influenzerebbero la costruzione.

L'applicazione, durante la fase di previsione, deve controllare, per ogni sensore, che la propria serie storica multi-variata non contenga valori non significativi ($MAX\text{-}VALUE$) negli istanti temporali coinvolti nella previsione. Nel caso in cui tale riscontro sia positivo per quel sensore sarà effettuato forecast, in alternativa no. In base a tale assunzione, se per un sensore viene effettuato forecast una volta, sarà sempre possibile effettuarne altri.

Capitolo 3

VARForecaster 2.1/2.2

In questo capitolo sono descritte le versioni dell'algoritmo (descritto nel par. 2.4) progettate ed implementate in questo lavoro di tesi.

Si ricorda che (come descritto nel par. 1.1) l'algoritmo è stato esteso al fine di calcolare gli aggregati dei dati in tempo reale (media o mediana) piuttosto che ricorrere ad approssimazione degli stessi e utilizzare un meccanismo di smoothing esponenziale per determinare il modello di clustering sulla base dei valori geofisici attesi nella rete all'istante **t+1** piuttosto che sulla base dei valori osservati all'istante **t**. La motivazione è l'identificazione di una soluzione che realizza il miglior bilanciamento tra accuratezza e efficienza.

3.1 VARForecaster 2.1

L'albero è costruito in maniera incrementale (par. 2.4), ovvero, ad ogni nuovo snapshot di dati all'interno dello stream, l'algoritmo controlla che le condizioni di split in base alle quali è stato costruito il modello di clustering all'istante t-1 siano compatibili con i dati raccolti all'istante t.

A tale scopo, per ogni nuovo snapshot, l'albero è aggiornato eseguendo una fase di pruning e poi di ri-apprendimento dell'albero sulla base dei nuovi valori ricevuti.

Il pruning è un processo ricorsivo nel quale vengono analizzati i nodi di split, i quali sono potati se la loro condizione di split non porta ad una reale riduzione della varianza degli indicatori locali di autocorrelazione spaziale sui nuovi record.

Nel caso in cui venga applicata la potatura, il nodo di split viene sostituito con un nodo foglia, al cui interno ricadranno tutte le istanze precedentemente appartenenti al nodo di split.

La fase di ri-apprendimento dell'albero, al contrario di quella di pruning, prende in considerazione i nodi foglia, è infatti in questa fase che l'algoritmo VARForecaster tenta di apprendere nuovi sotto-alberi.

Secondo la versione 2.0 dell'algoritmo, nel momento in cui viene appreso un nuovo nodo, a questo viene assegnata, nella struttura dati che tiene traccia delle medie multi-variate assunte da ogni cluster (cioè la struttura dati *Feature Window*), la media precedente calcolata per il suo nodo padre (vedi par. 2.4).

Aggregazione in tempo reale

Al fine di migliorare la rappresentatività dei dati usati per l'apprendimento del modello VAR, si è pensato di ricostruire, per ogni nuovo nodo appreso, la media multi-variata ad esso associata, in modo tale da mantenere un'aggregato multi-variato calcolato sulla base dei reali valori assunti dalle istanze dei sensori che ricadono all'interno del cluster foglia.

Questo processo è realizzato all'interno della struttura dati *Node* e si sviluppa nei seguenti passi:

- Per ogni sensore appartenente al cluster foglia:
  - viene letta la serie storica multi-variata ad esso associato;
  - per ogni caratteristica (*Feature*) presente all'interno della serie:
    - per ogni istante temporale precedente a t, viene letto il valore corrispondente e lo si somma agli altri valori letti per la caratteristica in esame;
    - viene incrementato un contatore, il quale sarà necessario per definire quanti istanti temporali sono stati sommati.

Alla fine di tale procedimento, per ogni sensore, sono calcolati, in maniera separata, la somma ottenuta per ogni *Feature* ed il valore del contatore a quella caratteristica associata.

Questi valori rappresentano la media, per ogni *Feature*, delle rilevazioni effettuate fino all'istante t dall'insieme di sensori che ricadono all'interno del cluster foglia. Tali valori sono poi accodati all'insieme delle rilevazioni precedentemente effettuate dal sensore, per ogni singola *Feature*, nella struttura dati *Feature Window*.

Problematiche

Nel momento in cui si prova a definire un'aggregato multi-variato per un cluster si può incorrere in due inconvenienti:

- Sensori inattivi;
- Presenza di outliers;

Sono di seguito descritte le modalità secondo le quali è stato possibile risolvere entrambi i problemi.

Sensori inattivi

Nel calcolo della media, come specificato in precedenza, vengono considerate tutte le rilevazioni presenti all'interno della finestra temporale associata al sensore.

Se tale sensore in passato, anche per un singolo istante temporale t_x, non ha effettuato alcuna rilevazione, nella struttura dati *Feature Window* ad esso associata, per l'istante t_x, è salvato il valore massimo che Java permette per rappresentare una variabile di tipo Double (di seguito definito come *"MAX-VALUE"*).

Se si prendesse in considerazione tale valore nel calcolo della media, quest'ultima risulterebbe definitivamente influenzata. Per evitare ciò, nel momento in cui si calcola la media, si controlla che il valore rilevato non sia *"MAX-VALUE"*. Nel caso in cui si incorre in tale valore, il sensore incriminato, non viene considerato per il calcolo della media e il contatore non subisce alcuna modifica.

Presenza di outliers

Data la varietà di dati misurati e la distribuzione geografica dei sensori, è possibile che tra le rilevazioni effettuate possano esserci uno o più ouliers.

Un outlier è un valore anomalo e aberrante; un valore quindi chiaramente distante dalle altre osservazioni disponibili [14]

Data la struttura di questi dati, che sarebbero assolutamente fuorvianti in caso di aggregati costruiti tramite l'utilizzo della media, è stato messa a disposizione dell'utente, come parametro per il lancio del software, la possibilità di scegliere se calcolare il suddetto aggregato con l'utilizzo della mediana.

In statistica, in particolare in statistica descrittiva, data una distribuzione di un carattere quantitativo oppure qualitativo ordinabile (ovvero le cui modalità possano essere ordinate in base a qualche criterio), si definisce la **mediana** (o **valore mediano**) come il valore assunto dalle unità statistiche che si trovano nel mezzo della distribuzione.

Se si procede al riordinamento delle unità in base ai valori crescenti del carattere da esso detenuto, in sostanza la mediana bipartisce la distribuzione in due sotto-distribuzioni: la prima a sinistra della mediana (costituita dalla

metà delle unità la cui modalità è minore o uguale alla mediana) e la seconda a destra della mediana costituita dalla metà delle unità la cui modalità è maggiore o uguale alla mediana). Tecnicamente si afferma che la mediana è il valore per il quale la frequenza relativa cumulata vale (o supera) 0,5.

Per calcolare la mediana di n dati [15]

- si ordinano gli n dati in ordine crescente;
- se il numero di dati è dispari la mediana corrisponde al valore centrale, ovvero al valore che occupa la posizione $\frac{n+1}{2}$;
- se il numero n di dati è pari, la mediana è stimata utilizzando i due valori che occupano la posizione $\frac{n}{2}$ e $(\frac{n}{2} + 1)$

A livello implementativo, la differenza fondamentale tra il calcolo degli aggregati multi-variati tramite media rispetto a farlo con la varianza, risiede nell'utilizzo di una struttura dati in cui vengono memorizzati i valori di ogni serie uni-variata per istante temporale t.

Da questa struttura si va poi a leggere il valore centrale, ovvero quello corrispondente alla mediana, per poi continuare ad operare come nel caso della media. Nel caso in cui invece il numero n di dati sia dispari, si provvede alla lettura dei due valori centrali e all'utilizzo della media per il calcolo del valore mediano.

Caso limite

Esiste, in maniera remota, la possibilità che tutti i sensori che ricadono in un cluster foglia, non abbiano mai effettuato alcuna rilevazione, fino all'istante temporale t in esame.

Per la gestione di questa casistica limite si è preferito quindi accodare un valore *"MAX-VALUE"*, come quello presente in caso di sensore inattivo.

Così facendo è in seguito catturata e gestita un'eccezione che non permette il forecast.

Conseguenze

A causa della rielaborazione degli aggregati multi-variati in caso di apprendimento di un nuovo nodo, le serie storiche di ciascun cluster foglia risultano essere ora reali, ovvero adattate alle istanze che realmente ricadono all'interno di tale cluster. Come conseguenza primaria si ha quindi, che la costruzione del modello VAR rispecchierà in maniera maggiore la situazione reale del cluster.

La possibilità di scegliere se costruire gli aggregati con la media oppure con la varianza verrà data all'utente utilizzatore del software grazie ad una parametrizzazione della stessa.

Nella prossima sezione si riportano i diagrammi di classe e di sequenza relativi alla progettazione di VARForecaster 2.1.

3.1.1 Progettazione

In questa sezione sono illustrate, tramite diagrammi UML (Unified Modeling Language) le variazioni progettate in questo lavoro di tesi dell'algoritmo VARForeacaster 2.1. Precisamente sono presenti: diagramma dei package, diagrammi delle classi e diagrammi di sequenza. Ognuno di essi è seguito da una descrizione.

Diagramma dei package

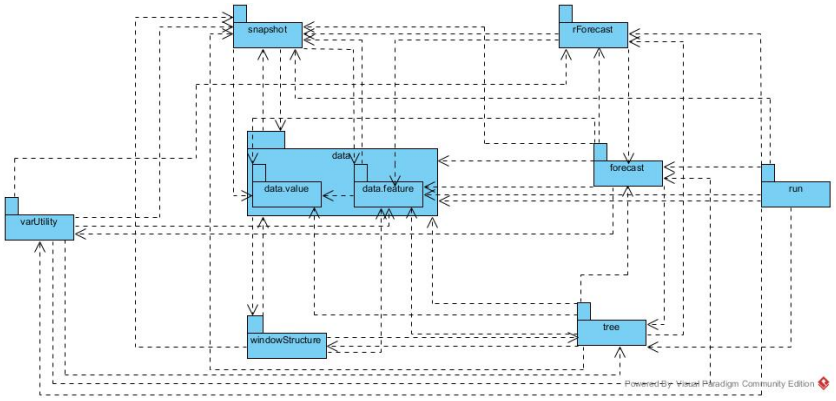

Figura 3.1: RealAggregate Package Diagram

Nella Fig. 3.1 sono descritti le principali relazioni che intercorrono tra i package presenti all'interno del software, ovvero come un package è in relazione con un altro.

A tali relazioni non è stata apportata alcuna modifica.

Diagrammi delle classi

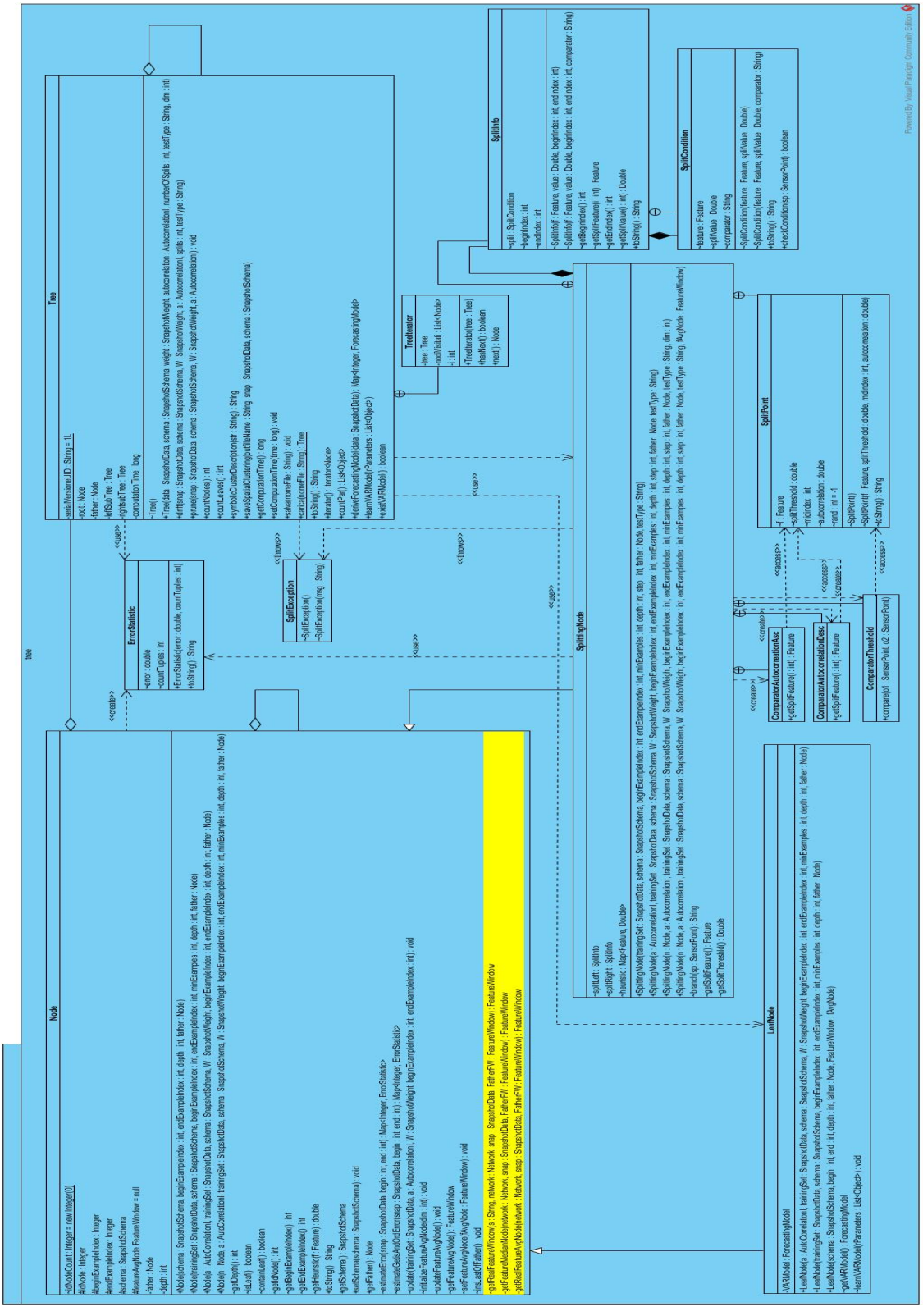

Figura 3.2: RealAggregate Class Diagram

Come è possibile notare, in Fig. 3.2, ci sono dei metodi in giallo. Questi sono i metodi aggiunti rispetto alla versione VARForecaster 2.0.

Tali metodi hanno la responsabilità di calcolare gli aggregati reali e sono utilizzati vicendevolmente rispetto al parametro ricevuto in input dall'utente. Precisamente:

- getRealFeatureWindow: è il metodo che ha la responsabilità di leggere il parametro che determina il tipo di funzione da applicare per calcolare l'aggregato (media o mediana), di chiamare il metodo corretto e restituire la struttura contenente i dati aggregati aggiornati;
- getFeatureMedianNode: è il metodo che ha il compito di calcolare l'aggregato reale con la funzione *"mediana"*;
- getRealFeatureAvgNode: è il metodo che ha la responsabilità di calcolare l'aggregare reale con la funzione *"media"*.

Tali metodi sono utilizzati, come detto nel par. 3.1, nella fase di apprendimento incrementale del clustering, quindi per l'esattezza nel metodo privato *learnTreeOnPrexistModel* nella classe *Tree*.

Queste modifiche sono esplicitate chiaramente, nella seguente sotto-sezione, dai diagrammi di sequenza.

Diagrammi di sequenza

3.2 VARForecaster 2.2

In questa versione del sistema VARForecaster 2.0 ai dati viene applicato, per la costruzione dell'albero (e per il suo apprendimento incrementale), un'algoritmo di smoothing esponenziale.

Si è pensato a questa modifica in quanto, adattare al meglio la costruzione dell'albero in previsione dei valori futuri assunti mediamente da ciascuna feature all'interno dei cluster, potrebbe migliorare l'accuratezza con un basso costo in termini di costi computazionali in tempo.

L'albero, nelle versioni 2.0 e 2.1, viene appreso su ogni snapshot ricevuto in input all'istante temporale t mentre in questa versione è appreso sulla previsione effettuata sui dati avuti nell'istante *t-1* tramite l'utilizzo della tecnica di smoothing esponenziale.

Per effettuare tale operazioni si sono rese necessarie alcune modifiche alla struttura del sistema che possono essere riassunte in 2 fasi:

(1) creazione dello snapshot previsto attraverso lo smoothing esponenziale (in seguito def. *snapSmoot*);

(2) apprendimento incrementale del CT;

Nei successivi paragrafi sono descritte entrambe le fasi con particolare attenzione alle modifiche effettuate.

Creazione dello *snapSmoot*

La fase di creazione dello *snapSmoot* avviene ogni qualvolta si processa un nuovo snapshot. Tale fase si preoccupa di leggere lo snapshot dell'istante temporale t in esame, precedentemente ricevuto in input, e processarlo al fine di produrre uno snapshot di previsioni di quelli che potrebbero essere i valori assunti da ciascuna feature di ciascuna istanza presente al suo interno.

Questo processo si compone di due casistiche:

- si tratta del primo istante temporale contenente i valori rilevati dai sensori;
- è un istante temporale successivo al primo, quindi abbiamo già applicato lo smoothing in precedenza.

Nel caso in cui sia stia analizzando il primo istante temporale, il sistema si preoccupa di clonare tutti i sensori ricevuti in input nello *snapSmoot*. Questo è dovuto al fatto che il passo base per il calcolo dello smoothing esponenziale è ottenuto con la copia dei dati ricevuti in ingresso.

Nel caso in cui invece si stia analizzando un istante temporale successivo al primo, il sistema ha la responsabilità di trovare, tra i sensori presenti nello *snapSmoot* di *t-1*, i sensori che nell'istante temporale t hanno effettivamente effettuato una rilevazione e combinarli applicando il metodo di *smoothing esponenziale di Brown*.

Sensori inattivi nella creazione dello *snapSmoot*

Come nel caso della versione VARForecaster 2.1, i sensori inattivi rappresentano un eventuale problematica da gestire.

Come precedentemente specificato, il sistema deve trovare tra i sensori presenti nello *snapSmoot* di *t-1* quelli che hanno effettivamente rilevato dei dati nell'istante temporale t in considerazione. Ciò implica due conseguenze:

(1) potrebbero esserci sensori presenti all'interno di *snapSmoot* ma non presenti all'interno dello snapshot attuale;

(2) potrebbero esserci sensori presenti nello snapshot attuale ma non presenti nello *snapSmoot*.

Per la gestione di (1) è stato scelto di ignorare tali sensori in quanto il processo di smoothing, prendendo in considerazione gli istanti temporali precedenti a t avrebbe un valore mancante, il quale condizionerebbe tutto il meccanismo.

Per la gestione di (2), per lo stesso motivo appena citato, si è preferito far ripartire il processo di smoothing dal "passo base" e quindi di clonare il sensore reale.

Apprendimento incrementale del *CT*

In questa fase di progetto si è fatta molta attenzione a distinguere due processi:

- aggiornamento dei dati relativi ai cluster
- costruzione dell'albero

Per aggiornamento dei dati relativi ai cluster si intende quella fase in cui vengono aggiornati, per ogni nodo, relativamente al cluster a cui fanno riferimento, i valori delle feature di ogni sensore. Questo vuol dire che i dati presenti in ogni nodo sono riferiti ai valori realmente rilevati dai sensori che ricadono in quel cluster e non ai valori predetti con lo smoothing.

La costruzione dell'albero invece avviene sulla base dei valori predetti con lo smoothing. Ciò indica che l'albero sarà adattato ai valori previsti per quel gruppo di sensori che ricade nel nodo in oggetto.

Nella fase di crescita incrementale dell'albero questo vuol dire che non soltanto la fase di apprendimento ha subito modifiche ma anche quella di potatura.

In questa fase si applica l'operatore di pruning sulla base dei valori predetti per lo snapshot *t-1* presenti in *snapSmoot* in quanto l'albero è stato appreso sulla base dei valori presenti in esso.

Tale processo avviene facendo attenzione, nella fase di aggiornamento dei dati in base ai valori ricevuti dal nuovo snapshot, a mantenere nel nodo i valori associati alle reali rilevazioni ottenute dai sensori che ricadono in quel nodo, mentre per il computo dell'autocorrelazione spaziale e per le decisioni di pruning, vengono utilizzati i valori presenti in *snapSmoot*.

La fase di apprendimento, detta anche *drift*, è la parte maggiormente modificata. Questo avviene perché le decisioni di split devono essere apprese in base allo *snapSmoot*, di conseguenza l'applicazione deve preoccuparsi di lavorare parallelamente su entrambi gli snapshot (*snapSmoot* e lo snapshot contenente i valori realmente rilevati).

Metodo scelto e parametrizzazione

Il metodo di smoothing utilizzato in questo progetto di tesi è il *metodo di Brown*. Si riporta la l'espressione matematica di tale metodo:

Passo ricorsivo

$$Y_t = \alpha\ x_t + (1 - \alpha)\ Y_{t-1}$$

Passo base

$$Y_t = x_t$$

dove, in questo progetto di tesi:

- x_t è lo snapshot contenente i valori realmente rilevati dai sensori;
- Y_{t-1} è lo *snapSmoot* ottenuto dall'applicazione della funzione di smoothing esponenziale sul precedente snapshot contente i valori reali (x_{t-1});
- α è il parametro di smoothing.

La scelta del parametro di smoothing è arbitraria e affidata all'utente, ma per non adattare troppo i dati previsti a quelli reali o a quelli precedente previsti è consigliabile mantenersi entro $0.4 \leq \alpha \leq 0.6$.

3.2.1 Diagrammi di classe

3.2.2 Diagrammi di sequenza

Capitolo 4

Analisi sperimentale

4.1 Obiettivi e metriche

4.2 Dati

4.3 Risultati

4.3.1 EcoTexas

4.3.2 UnitedStatesPacific

Capitolo 5

Conclusioni

Capitolo 6

Appendice

6.1 Guida per l'esecuzione di VARForecaster 2.1/2.2

Capitolo 7

Bibliografia

Bibliografia

[1] D. Mastropasqua. *Apprendimento di cluster spazio-temporali per l'apprendimento di un modello VAR*, 2016: Tesi di laurea in Algoritmi e Strutture Dati, Uniba.

[2] G. Data, P. Mariani. *Market Access nel settore healthcare – Strategie, attori, attività e processi*, 2015: FrancoAngeli Editori.

[3] C. Alexander. *Market Risk Analysis, Value at Risk Models*, 2009: Wiley.

[4] N. Fanizzi. *Corso di Apprendimento Automatico*, 2009: Laurea Magistrale in Informatica, Uniba.

[5] H. Arsham. *Forecasting by Smoothing Techniques*, 2015.

[6] M. Brett. *Corso di Apprendimento Automatico*, 2013 : MRC Cognition and Brain Sciences Unit.

[7] R.J. Hyndman, G. Athanasopoulos. *Forecasting: principles and practice*, 2013.

[8] G. Righini. *Corso di Logistica*, 2013: Laurea Magistrale in Informatica, Unimi.

[9] C.A. Sims. *Macroeconomics and Reality*, 1980.

[10] J.D. Hamilton. *Time Series Analysis*, 1994: Princeton University Press.

[11] D. Buono. *Analisi econometrica dinamica del settore Agricoltura*, 2007: Tesi di dottorato in Scienze Economiche, Unina.

[12] *https://www.r-project.org/about.html*.

[13] V. Fanelli. *Uso di modelli VAR nella previsione di modelli di regressione in una rete di sensori multi-variati*, 2015: Tesi di laurea in Metodi Avanzati di Programmazione, Uniba.

[14] *https://it.wikipedia.org/wiki/Outlier*.

[15] M. Ross Sheldon. *Introduzione alla statistica*, 2014, Maggioli Editore.

[16] L. Bing. *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*, 2011, par. 4.1.

[17] L. Bing. *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*, 2011, par. 4.2.

[18] L. Bing. *Web Data Mining Exploring Hyperlinks, Contents, and Usage Data*, 2011, par. 4.3.

[19] A. Appice, D. Malerba *Leveraging the power of local spatial autocorrelation in geophysical interpolative clustering, Data Mining and Knowledge Discovery*, 2014, cap. 2.

[20] *https://cran.r-project.org/web/packages/vars/vars.pdf*.

Ringraziamenti