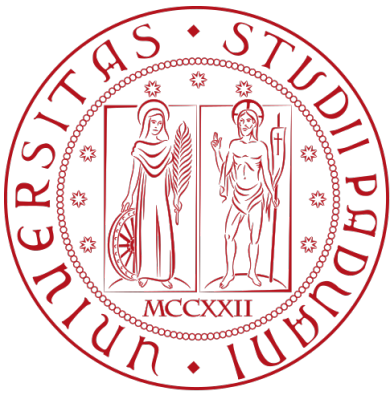
Università degli Studi di Padova

Dipartimento di Scienze Statistiche

Corso di Laurea Magistrale in
Scienze Statistiche



Modelli statistici per dati meteorologici

Relatore: **Prof. Mauro Bernardi**
Dipartimento di Scienze Statistiche

Laureando: **Filippo Scalabrin**
Matricola: **2071953**

Anno Accademico 2023/2024

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Fin dai tempi degli antichi Greci, la meteorologia è sempre stata al centro dell'attenzione degli scienziati di ogni civiltà, visto il potenziale degli eventi atmosferici di condizionare, talvolta pesantemente, la vita dell'Uomo sulla Terra. Forti grandinate e trombe d'aria, assieme a siccità e ondate di calore, sono alcune tra le calamità che da sempre — ma in particolare negli ultimi anni — determinano criticità idrogeologiche, aridità dei terreni e danni a cose o a persone. Secondo la European Environment Agency (EEA), tra il 1980 e il 2023 gli eventi estremi di questo tipo hanno causato perdite economiche stimate in 650 miliardi di euro nei Paesi dell'Unione Europea (Casadei & Finizio, 2024). Saper modellare nel modo più idoneo dati meteorologici — a maggior ragione, in un'epoca caratterizzata dal rapido manifestarsi dei cambiamenti climatici — consente di poter interpretare l'andamento temporale dei fenomeni atmosferici, formulare delle previsioni attendibili e, eventualmente, di pianificare interventi di prevenzione ambientale o di messa in sicurezza.

A partire dal 2014, anno di nascita del servizio tematico Copernicus Climate Change (abbreviato in C3S), l'Unione Europea si è proposta di mettere a disposizione di chiunque una variegata serie di informazioni affidabili sullo stato attuale e pregresso del clima. A rendere concreto questo progetto di sensibilizzazione è stato il lavoro condotto dallo European Centre for Medium-Range Weather Forecast (ECMWF), che ha portato alla creazione delle cosiddette "rianalisi". Con questo termine si definisce il quadro più completo possibile delle condizioni meteorologiche e climatologiche del passato. I dati necessari alla formulazione di una rianalisi derivano sia dalle misurazioni realmente effettuate tramite appositi apparati, quali centraline, radar e pluviometri, sia dalle stime storiche ottenute mediante la tecnica NWP (Numerical Weather Prediction), che fa perno sulle leggi fisiche della fluido-

dinamica e della termodinamica (Ben Bouallègue et al., 2024). Naturalmente, tali stime presentano un certo grado di incertezza: i modelli numerici possono solo fornire una rappresentazione più o meno accurata dei veri processi fisici che governano il meteo sulla Terra, e l'imprecisione nel calcolo a posteriori dei parametri ambientali cresce man mano che si va indietro nel tempo (Muñoz Sabater, 2019).

La quinta versione delle rianalisi dell'ECMWF disponibili in rete, conosciute con l'acronimo di ERA5 (ECMWF Reanalysis v5), consiste dunque in una vastissima collezione di dati meteorologici relativi al periodo temporale che va dal 1940 al presente, organizzati in griglie territoriali equamente spaziate. Le variabili di ERA5 fanno riferimento a grandezze rilevate in atmosfera, nelle acque o sulla superficie terrestre. Sono usate dall'ECMWF stesso a supporto di modelli statistici atti ad integrare e migliorare le tecniche di previsione fondate sulle leggi della Fisica, e rendono agevole lo studio dell'andamento delle condizioni meteorologiche medie di una regione in un determinato periodo di tempo.

L'obiettivo della presente tesi è quello di fornire alcuni strumenti statistici adatti a padroneggiare dati meteorologici, a partire dall'analisi dei dati di ERA5. L'interesse primario, ma non esclusivo, è rivolto verso un'ampia classe di modelli che permette di cogliere la struttura di dipendenza insita in osservazioni riferite a punti nello spazio e nel tempo: quella degli hidden Markov models (o "modelli di Markov nascosti"). La tesi è organizzata come segue. Nel Capitolo 1 spiegheremo com'è avvenuta la raccolta dei dati, elencando le località a cui questi si riferiscono e specificando in dettaglio le variabili coinvolte. Nel Capitolo 2 mostreremo i risultati salienti dell'analisi esplorativa condotta sui dataset e di un insieme di procedure di clustering volte ad accomunare, per caratteristiche climatiche, le località in esame. Il Capitolo 3 è dedicato alla discussione degli hidden Markov models e delle loro estensioni. Dopo la descrizione dei modelli da un punto di vista teorico, troveranno spazio alcune loro applicazioni ai dati di ERA5, corredate da diversi commenti. Nel Capitolo 4 esploreremo, con un taglio più descrittivo, altri interessanti fronti della modellazione statistica di dati meteorologici. Infine, nelle Conclusioni discuteremo dei margini di miglioramento del lavoro svolto, riassunto nella sua globalità.

L'estrazione dei dati e la realizzazione di analisi e grafici sono state condotte per la maggior parte mediante software R (versione 4.3.3); solo per ricavare una variabile aggiuntiva è stato usato anche Python (versione 3.8). Al termine delle Conclusioni è consultabile un'Appendice con gli estratti principali del codice prodotto.

Capitolo 1

ERA5 e i dataset

In questo capitolo, effettueremo in primis una panoramica sulla natura del database dal quale è stato possibile ricavare le variabili meteorologiche utilizzate per le analisi, ERA5, illustrando il meccanismo di estrazione dei dati operato dall'API scelta per lo scopo. In seguito, elencheremo in modo esaustivo tutte le località prese in considerazione; infine, rappresenteremo in tabella le variabili ricavate, commentandole brevemente.

1.1 Meccanismo di estrazione dei dati

Le rianalisi di ERA5 sono organizzate per griglie territoriali regolari, come anticipato nell'Introduzione, e sono scaricabili nel formato GRIB (General Regularly distributed Information in Binary form), definito come standard per dati meteorologici dall'Organizzazione Meteorologica Mondiale (WMO). Per capire nel dettaglio la struttura del database, immaginiamo di visualizzare una qualsiasi area geografica e di ricoprirla con una griglia di punti equamente distanziati tra loro, seguendo il suggerimento della stilizzazione di Figura 1.1; a questa griglia, sovrapponiamone tante quante sono gli istanti temporali disponibili. Ad ognuno dei punti di rilevazione definiti dalle griglie sovrapposte, ECMWF ha associato le serie storiche di un gran numero di variabili meteorologiche e di qualità dell'aria (Hersbach et al., 2018). Tra i dati che compongono queste serie storiche figurano sia valori realmente acquisiti dagli apparati meteorologici nel passato, sia valori interpolati.

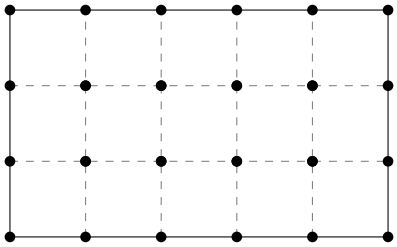


Figura 1.1: Rappresentazione grafica della struttura del database ERA5 (ECMWF Reanalysis v5).

Un modo per guadagnare l'accesso ai dati di ERA5 consiste dunque nel visitare il sito web del Copernicus Climate Change Service e, previa registrazione al portale, nello scaricare un file GRIB specificando i limiti geografici dell'area rettangolare d'interesse. Una valida alternativa al download diretto del file GRIB è l'API del sito Open-Meteo.com (Zippenfenig, 2024), accessibile tramite il pacchetto R `openmeteo` (nella sua versione 0.2.4 al momento del lavoro). Operativamente, con le funzioni di `openmeteo` si possono ottenere i dati di interesse di una qualsiasi città, inserita in input sotto forma di stringa o tramite una coppia di coordinate latitudine-longitudine. Nel primo caso, occorre essere consapevoli del fatto che vi possono essere, nel mondo, città con lo stesso nome (ad esempio, esiste un paese chiamato Milano a nord-est di Austin, in Texas) e l'API può restituire dati non desiderati. Pertanto, per chiunque volesse usare `openmeteo`, il consiglio è di immettere i nomi di città in lingua inglese e di controllare accuratamente ex post i valori delle variabili ricavate. Nonostante questo aspetto a cui prestare attenzione, la chiamata alle funzioni di `openmeteo` restituisce i dati in una familiare forma matriciale, e consente di specificare il tipo di granularità ricercata (oraria o giornaliera). Al contrario, l'apertura e l'ispezione di un file GRIB possono risultare difficoltose senza ricorrere ad appropriati software di meteorologia; in più, con il download diretto, la granularità dei dati è vincolata ad essere oraria.

Vediamo in che modo `openmeteo` riesce a restituire dati relativi a precise località, a partire dall'originale formato GRIB. Supponiamo che l'area geografica stilizzata in Figura 1.1, presentata ancora in Figura 1.2, corrisponda ad una porzione di territorio del Baden-Württemberg, in Germania, e che l'interesse sia quello di acquisire i dati meteorologici di Stoccarda, rappresentata dal puntino blu. L'API restituisce semplicemente i dati di ERA5 corrispondenti al punto più vicino nello spazio alla città stessa, che nel caso di Figura 1.2 risulta essere quello di coordinate fittizie $(3, 2)$. Questo può portare a leggere incongruenze tra dati "attesi" e dati in output qualora la località di interesse non sia nei pressi di uno dei punti della griglia, specialmente se si trova in montagna, dove spostamenti anche piccoli nello spazio possono portare a diversi valori meteorologici a causa delle differenze di altitudine.

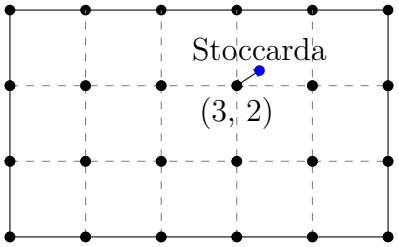


Figura 1.2: Matching tra città specificata in input e dato più vicino disponibile effettuato dall'API di Open-Meteo.

1.1.1 Limiti e restrizioni dell'API

Purtroppo, con `openmeteo` non è possibile richiedere l'estrazione di ognuna delle variabili presenti in ERA5 e visibili nel sito web del C3S, anche se comunque — come avremo modo di constatare in seguito — tutte quelle più interpretabili dai non esperti del settore (riguardanti temperatura, vento, precipitazioni, ecc.) sono disponibili. Oltretutto, pur essendo gratuita, l'API presenta dei limiti di utilizzo che contingentano i download effettuati ogni minuto, ora e giorno, legati anche ai limiti imposti dal Copernicus Climate Change Service sul download diretto dei dati grezzi in formato GRIB. I suddetti limiti vengono raggiunti presto se l'intento è di estrarre tutte le variabili accessibili per molte località e periodi di tempo ampi; ragion per cui, per costruire i dataset che verranno presentati nel paragrafo successivo, raccolta ed organizzazione dei dati hanno richiesto molti giorni di lavoro manuale.

Infine, la risoluzione dei dati, ovvero la distanza spaziale dei punti tra di loro (nelle rappresentazioni di Figura 1.1 e 1.2, la lunghezza del lato dei quadratini definiti dai punti neri) è pari a 31 km. Non è quindi sensato raccogliere i dati a livello di piccoli paesi molto ravvicinati tra di loro, perché sarebbero probabilmente sempre gli stessi. ECMWF sta lavorando ad una nuova versione delle rianalisi, ERA6, che dovrebbe avere una risoluzione di 18 km (Hersbach et al., 2022).

1.2 Costruzione dei dataset per l'analisi

Per la presente dissertazione abbiamo costruito, grazie all'API Open-Meteo, due dataset. Il primo contiene dati a granularità giornaliera relativi al periodo 2010–2023 e riferiti a 225 città europee ed africane, localizzate su mappa in Figura 1.3 ed elencate in Tabella 1.1. La scelta delle città non è stata guidata da particolari criteri. Inizialmente, l'intento era di concentrarsi solamente sui capoluoghi di provincia italiani, motivo per cui tutti compaiono nel dataset; in seguito, però, abbiamo deciso di estendere la raccolta dei dati a città estere.

Tabella 1.1: Elenco delle città per le quali si dispone di dati giornalieri.

Stato	Città
Albania	Corizza, Scutari, Tirana, Valona
Algeria	Algeri, Laghouat
Austria	Graz, Innsbruck, Salisburgo, Vienna, Villaco
Belgio	Anversa, Bruges, Charleroi, Liegi
Bosnia-Erzegovina	Banja Luka, Mostar, Sarajevo, Tuzla
Bulgaria	Plovdiv, Sofia
Croazia	Bencovazzo, Daruvar, Fiume, Osijek, Pago, Pola, Spalato, Umago, Zagabria
Francia	Ajaccio, Amiens, Bordeaux, Brest, Digione, Grenoble, Lione, Marsiglia, Metz, Montpellier, Nantes, Nizza, Parigi, Strasburgo, Tolosa
Germania	Amburgo, Berlino, Dortmund, Dresda, Francoforte sul Meno, Hannover, Monaco di Baviera, Norimberga, Stoccarda
Grecia	Alessandropoli, Atene, Candia, Corfù, Salonicco, Santorini, Zante
Italia	Tutti i capoluoghi di provincia
Libia	Bengasi, Tripoli
Lussemburgo	Lussemburgo
Macedonia	Bitola, Skopje, Strumica
Marocco	Fes, Marrakech
Montenegro	Berane, Budva, Podgorica
Paesi Bassi	Amsterdam, Eindhoven, Groninga, Rotterdam, Zwolle
Polonia	Cracovia, Wroclaw
Portogallo	Lagos, Lisbona, Porto
Repubblica Ceca	Brno, Praga
Romania	Bucarest, Cluj-Napoca
Serbia	Belgrado, Kragujevac, Leskovac, Novi Sad, Subotica
Slovacchia	Bratislava, Košice, Zvolen
Slovenia	Bled, Lubiana, Marburgo, Novo Mesto
Spagna	Almería, Barcellona, Bilbao, La Coruña, Madrid, Malaga, Palma de Maiorca, Saragozza, Siviglia, Valencia
Svizzera	Berna, Davos, Losanna, Sion, Zurigo
Tunisia	Sfax, Tunisi
Ungheria	Budapest, Debrecen, Pécs

Tabella 1.2: Elenco dei comuni del Veneto per le quali si dispone di dati orari.

Provincia	Comuni
Belluno	Agordo, Belluno, Cortina d'Ampezzo, Feltre, Mel, Pieve di Cadore
Padova	Abano Terme, Cittadella, Monselice, Montagnana, Padova, Piombino Dese, Piove di Sacco, Stanghella
Rovigo	Adria, Badia Polesine, Castelmassa, Porto Tolle, Rovigo, Stienta
Treviso	Castelfranco Veneto, Conegliano, Montebelluna, Oderzo, Possagno, Treviso, Spresiano, Vittorio Veneto
Venezia	Bibione, Cavarzere, Chioggia, Jesolo, Mirano, Portogruaro, San Donà di Piave, Venezia
Verona	Albaredo d'Adige, Bosco Chiesanuova, Cerea, Isola della Scala, Malcesine, Peschiera del Garda, Soave, Spiazzi, Tregnago, Verona
Vicenza	Asiago, Arzignano, Barbarano Vicentino, Bassano del Grappa, Lonigo, Noventa Vicentina, Valdagno, Vicenza

Il secondo dataset contiene dati a frequenza oraria relativi all'anno 2023 per i 54 comuni veneti di Tabella 1.2, rappresentati su mappa in Figura 1.4. A differenza del primo, questo dataset non contiene solo misurazioni meteoclimatiche, bensì anche variabili sulla qualità dell'aria in termini di concentrazione di pollini e inquinanti, sempre tratte dalle rianalisi di ERA5. Al momento della raccolta dei dati, le variabili sulla qualità dell'aria erano disponibili solo a partire dal luglio 2022; motivo per il quale il dataset a granularità giornaliera, descritto in precedenza, ne è privo.

Anche in questo caso, la scelta dei comuni è stata sostanzialmente arbitraria. Abbiamo cercato di coprire in maniera più o meno omogenea tutto il territorio di interesse, senza tenere conto di fattori quali altitudine o appartenenza a particolari microclimi (cioè zone geografiche ristrette dotate, per questioni ambientali, di parametri atmosferici peculiari e differenti in modo caratteristico da quelli delle zone circostanti). Per quanto messo in luce al termine del Paragrafo 1.1.1, particolare attenzione è stata rivolta, soprattutto al momento della costruzione del dataset a granularità temporale oraria, al non scegliere città troppo vicine geograficamente, per le quali i dati di ERA5 sarebbero stati i medesimi.

Le variabili meteorologiche a disposizione per i due dataset sono elencate e descritte nelle Tabelle 1.3 (dati giornalieri) e 1.4 (dati orari), mentre quelle sulla qualità dell'aria, tutte di tipo numerico, sono consultabili in Tabella 1.5.

Figura 1.3: Posizione geografica delle città di cui si dispone di dati giornalieri.

Figura 1.4: Posizione geografica dei 54 comuni del Veneto di cui si dispone di dati orari.

Tabella 1.3: Descrizione delle variabili relative ai dati giornalieri.

#	Nome	Unità	Descrizione
1	id		Regione d'Italia o nazione di appartenenza.
2	date		Data ($YYYY - MM - DD$).
3	temperature_2m_max	°C	Temperatura massima a 2 m s.l.m.
4	temperature_2m_min	°C	Temperatura minima a 2 m s.l.m.
5	apparent_temperature_max	°C	Temperatura massima percepita.
6	apparent_temperature_min	°C	Temperatura minima percepita.
7	precipitation_sum	mm	Somma delle precipitazioni giornaliere.
8	precipitation_hours		Numero di ore con pioggia.
9	weather_code		La peggior condizione meteo giornaliera.
10	sunrise		Data e ora dell'alba.
11	sunset		Data e ora del tramonto.
12	wind_speed_10m_max	km/h	Velocità del vento a 10 m s.l.m.
13	wind_gust_10m_max	km/h	Velocità massima delle raffiche di vento a 10 m s.l.m.
14	wind_direction_10m_dominant	°	Direzione predominante del vento nel corso della giornata.
15	shortwave_radiation_sum	MJ/m^2	Radiazione solare giornaliera.
16	et0_fao_evapotranspiration	mm	Evapotraspirazione (evaporazione dell'acqua dal suolo e traspirazione dalle piante).
17	rain_sum	mm	Quantità giornaliera di pioggia caduta.
18	snowfall_sum	cm	Quantità di neve caduta.
19	sunshine_duration	s	Durata del soleggiamento escludendo alba e tramonto.
20	sunshine_duration	s	Durata del soleggiamento.
21	city		Nome della città.
22	lat	DD	Latitudine.
23	long	DD	Longitudine.

Puntualizziamo che i due dataset, costruiti aggiungendo i dataframe di ogni città uno sotto l'altro, risultano essere in quello che viene definito "formato lungo", tipico dei dati longitudinali (i quali derivano dall'osservazione di differenti variabili in diverse occasioni temporali). A loro volta, i dati di ogni città sono ordinati temporalmente a partire dalla misurazione meno recente.

Tabella 1.4: Descrizione delle variabili atmosferiche relative ai dati orari.

#	Nome	Unità	Descrizione
1	datetime		Data e ora della misurazione.
2	temperature_2m	°C	Temperatura dell'aria a 2 m s.l.m.
3	relative_humidity_2m	%	Umidità percentuale a 2 m s.l.m.
4	dew_point_2m	°C	Punto di rugiada a 2 m s.l.m.
5	apparent_temperature	°C	Temperatura percepita.
6	pressure_msl	hPa	Pressione atmosferica.
7	cloud_cover	%	Copertura nuvolosa.
8	cloud_cover_low	%	Copertura nuvolosa fino a 2 km di altitudine.
9	cloud_cover_mid	%	Copertura nuvolosa da 2 a 6 km di altitudine.
10	cloud_cover_high	%	Copertura nuvolosa a più di 6 km di altitudine.
11	wind_speed_10m	km/h	Velocità del vento a 10 m di altitudine.
12	wind_speed_100m	km/h	Velocità del vento a 10 m di altitudine.
13	wind_direction_10m	°	Direzione predominante del vento a 10 m di altitudine.
14	wind_direction_100m	°	Direzione predominante del vento a 100 m di altitudine.
15	wind_gusts_10m	km/h	Velocità massima del vento a 10 m s.l.m.
16	shortwave_radiation	W/m^2	Radiazione solare ad onde corte per l'ora precedente.
17	direct_radiation	W/m^2	Radiazione solare diretta per l'ora precedente su piano orizzontale.
18	direct_normal_irradiance	W/m^2	Radiazione solare diretta per l'ora precedente perpendicolarmente al Sole.
19	diffuse_radiation	W/m^2	Radiazione solare diffusa per l'ora precedente.
20	vapour_pressure_deficit	kPa	Differenza tra umidità registrata e umidità nel caso di aria satura.
21	et0_fao_evapotranspiration	mm	Evapotraspirazione.
22	precipitation	mm	Accumulo di precipitazioni per l'ora precedente.
23	rain	mm	Accumulo di pioggia per l'ora precedente.
24	weather_code		Condizione meteorologica rilevata secondo codice WMO.
25	snowfall	cm	Accumulo nevoso dell'ora precedente.
26	soil_temperature_0_to_7cm	°C	Temperatura del suolo da 0 a 7 cm di profondità.
27	soil_temperature_7_to_28cm	°C	Temperatura del suolo da 7 a 28 cm di profondità.
28	soil_temperature_28_to_100cm	°C	Temperatura del suolo da 28 a 100 cm di profondità.
29	soil_temperature_100_to_255cm	°C	Temperatura del suolo da 100 a 255 cm di profondità.
30	soil_moisture_0_to_7cm	m^3/m^3	Contenuto volumetrico medio di acqua nel suolo da 0 a 7 cm di profondità.
31	soil_moisture_7_to_28cm	m^3/m^3	Contenuto volumetrico medio di acqua nel suolo da 7 a 28 cm di profondità.
32	soil_moisture_28_to_100cm	m^3/m^3	Contenuto volumetrico medio di acqua nel suolo da 28 a 100 cm di profondità.
33	soil_moisture_100_to_255cm	m^3/m^3	Contenuto volumetrico medio di acqua nel suolo da 100 a 255 cm di profondità.
34	city		Città o comune.
35	lat	DD	Latitudine.
36	long	DD	Longitudine.

Tabella 1.5: Descrizione delle variabili relative alla qualità dell'aria, presenti nel dataset a granularità oraria.

#	Nome	Unità	Descrizione
1	pm10	$\mu g/m^3$	Concentrazione di PM10 a 10 m s.l.m.
2	pm2_5	$\mu g/m^3$	Concentrazione di PM2.5 a 10 m s.l.m.
3	alder_pollen	Grani/m^3	Concentrazione di polline di ontano.
4	birch_pollen	Grani/m^3	Concentrazione di polline di betulla.
5	grass_pollen	Grani/m^3	Concentrazione di polline di graminacee.
6	mugwort_pollen	Grani/m^3	Concentrazione di polline di artemisia.
7	olive_pollen	Grani/m^3	Concentrazione di polline di olivo.
8	ragweed_pollen	Grani/m^3	Concentrazione di polline di ambrosia.
9	uv_index		Indice UV tenendo conto della copertura nuvolosa.
10	uv_index_clear_sky		Indice UV con cielo terso.
11	dust	$\mu g/m^3$	Concentrazione di sabbia desertica in sospensione.
12	aerosol_optical_depth		Misura della riduzione della luce solare a causa di polveri.
13	carbon_monoxide	$\mu g/m^3$	Concentrazione di CO (monossido di carbonio) a 10 m s.l.m.
14	nitrogen_dioxide	$\mu g/m^3$	Concentrazione di NO_2 (biossido di azoto) a 10 m s.l.m.
15	sulphur_dioxide	$\mu g/m^3$	Concentrazione di SO_2 (anidride solforosa) a 10 m s.l.m.
16	ozone	$\mu g/m^3$	Concentrazione di O_3 (ozono) a 10 m s.l.m.
17	ammonia	$\mu g/m^3$	Concentrazione di NH_3 (ammoniaca) nell'aria.
18	european_aqi		Indice complessivo di qualità dell'aria.

In Tabella 1.6 definiamo nel dettaglio i livelli della variabile categoriale weather_code, presente in entrambi i dataset, che indica — per mezzo di un codice derivante dalla nomenclatura WMO — la condizione meteorologica più estrema (ovvero, la peggiore) rilevata in un'ora o in un giorno. I codici WMO da 0 a 3 indicano assenza di precipitazioni; quelli dal 51 al 55 la cosiddetta "drizzle", o pioviggine — un tipo di precipitazione costituito da goccioline di pioggia di diametro inferiore a 0.5 mm. Rappresentano la pioggia vera e propria i codici dal 61 al 65, mentre per la neve vi sono quelli dal 71 al 75. Ad esempio, quindi, un giorno estivo prevalentemente sereno, ma con un veloce temporale di intensità moderata, viene etichettato con il codice 63.

Appare sorprendente che weather_code manchi di un livello dedicato alla nebbia, fenomeno comune soprattutto nelle aree pianeggianti in presenza di alta pressione. Non abbiamo trovato spiegazioni a riguardo nella documentazione di ERA5. Visto

Tabella 1.6: Livelli della variabile weather_code: codici WMO per le condizioni atmosferiche.

Codice	Descrizione		Codice	Descrizione	
0	Sereno		61	Pioggia debole	
1	Poco nuvoloso		63	Pioggia moderata	
2	Parzialmente nuvoloso		65	Pioggia forte	
3	Coperto		71	Neve debole	
51	Pioviggine debole		73	Neve moderata	
53	Pioviggine moderata		75	Neve forte	
55	Pioviggine forte				

che, in termini pratici, la nebbia è un insieme di nuvole a bassa quota, probabilmente la scelta dei creatori del database è stata quella di assimilare il fenomeno ad una condizione di cielo coperto (livello 3) o di pioviggine (livelli 51, 53, 55).

Capitolo 2

Analisi esplorative sui dataset

Prima di entrare nel cuore della dissertazione, illustreremo i risultati salienti di un variegato lavoro di analisi esplorativa effettuato su entrambi i dataset descritti al Capitolo 1. Porremo un occhio di riguardo sulle variabili relative alle precipitazioni e alla temperatura, vista la grande attenzione, anche mediatica, che al giorno d'oggi viene posta sulle conseguenze dei cambiamenti climatici in atto in termini di siccità e riscaldamento globale. Per il dataset a granularità oraria, inoltre, indagheremo le variazioni nella concentrazione degli inquinanti nel corso del 2023 e stileremo una graduatoria delle città migliori per qualità dell'aria.

2.1 Analisi esplorativa dei dati giornalieri

In Tabella 2.1 presentiamo, con riferimento al dataset a granularità giornaliera, una panoramica sui valori climatologici record registrati dal 2010 al 2023, tenendo in considerazione tutte le 225 città. Non molto sorprendentemente, la temperatura massima più elevata è stata registrata in una città africana, Tunisi, il 24 luglio 2023. Come vedremo anche nel Paragrafo 2.2.2, durante la terza decade del mese di luglio 2023 un anticiclone subtropicale di matrice africana aveva abbandonato il Nord Italia, esponendo quest'ultimo a forti temporali, e determinando — spostandosi verso Algeria, Baleari e Tunisia — numerosi record termici. È della città svizzera di Davos, posta ad un'altitudine di 1560 m, il record di temperatura minima più bassa (−28.6 °C nel dicembre 2010). Interessante è poi la statistica che coinvolge

Tabella 2.1: Statistiche climatologiche sul dataset con i dati giornalieri.

Record	Valore	Località	Data
Temperatura massima	48.5 °C	Tunisi (TUN)	24/07/2023
Temperatura minima	−28.6 °C	Davos (CH)	27/12/2010
Escursione termica	14.1 °C	Bilbao (SPA)	29 – 30/07/2020
Escursione termica giornaliera	26.2 °C	Rieti (ITA)	28/02/2018
Pioggia	188.4 mm	Imperia (ITA)	24/11/2016
Neve	67.41 cm	Sondrio (ITA)	04/04/2019
Raffica di vento	182.5 km/h	Reggio Calabria (ITA)	11/11/2019

Bilbao, capoluogo dei Paesi Baschi (Spagna settentrionale): tra il 29 e il 30 luglio 2020, la temperatura massima è passata da 41.2 °C a 27.1 °C. Questo crollo viene confermato da altre fonti di dati meteo storici come Weather Underground e da tutte le stazioni disponibili nei pressi di Bilbao. Per quanto riguarda l'escursione termica giornaliera, intesa come la differenza tra temperatura massima e minima, a Rieti (nel Lazio), il 28 febbraio 2016 la colonnina di mercurio è passata da 2.8 °C di massima a −23.4 °C di minima. Abbiamo verificato anche questo dato anomalo (vedere la sezione "openAmbiente" del sito della Regione Lazio); tuttavia, ipotizziamo che la rielaborazione fornita dall'API Open-Meteo per Rieti sia in realtà quella di un punto nello spazio più vicino al Monte Terminillo (rimandiamo al Capitolo 1 per i dettagli su come avviene l'estrazione dei dati). Gli ultimi tre record di Tabella 2.1 appartengono anch'essi a città italiane. Imperia, dove il 24 novembre 2016 si è verificata un'alluvione a causa delle piogge torrenziali, detiene quello dei millimetri di pioggia giornalieri. A Sondrio va il record sui centimetri di neve; ma analogamente a quanto visto per Rieti, probabilmente i dati restituiti dall'API si riferiscono ad un punto più vicino ad una vicina montagna rispetto alla città vera e propria. Reggio Calabria, infine, è stata sferzata da fortissime raffiche di vento nel novembre 2019.

2.1.1 Andamento della siccità

Con riferimento ai cambiamenti climatici, sentiamo spesso parlare della "tropicalizzazione del clima" delle aree geografiche appartenenti alla fascia temperata dell'emisfero boreale, tra cui quelle del bacino del Mediterraneo. Con questa espressione si identifica la scomparsa della proporzione "tipica" tra giorni di pioggia e senza precipitazioni, a vantaggio di lunghi periodi siccitosi intervallati da passaggi piovosi concentrati in periodi ricorrenti.

Per studiare il fenomeno, decidiamo nel seguito di usare la diffusa convenzione che considera come "piovoso" un giorno con precipitazioni superiori al millimetro nell'arco delle 24 ore, ricordando che che "un millimetro di pioggia" indica un litro

d'acqua piovana su un metro quadrato di terreno. Definiamo le variabili

$$\mathsf{S}_{i,j,k} = \begin{cases} 0 & \text{se } \mathsf{precipitation\_sum}_{i,j,k} > 1 \text{ mm} \\ 1 & \text{se } \mathsf{precipitation\_sum}_{i,j,k} \leq 1 \text{ mm} \end{cases}$$

e

$$\mathsf{SC}_{i,j,k} = \begin{cases} 0 & \text{se } \mathsf{S}_{i,j,k} = 0 \\ 1 + \mathsf{SC}_{i,j,k-1}, & \text{se } \mathsf{S}_{i,j,k} = 1 \end{cases},$$

con $i = 1, \ldots, m$, $j = 1, \ldots, T$ e $k = 1, \ldots, 365$ indici di località, anno e giorno dell'anno, nell'ordine. $\mathsf{S}_{i,j,k} = 1$ se per la città i, nel giorno k dell'anno j vi sono state precipitazioni complessivamente inferiori al millimetro; $\mathsf{SC}_{i,j,k}$ conta i giorni consecutivi di siccità e si azzera quando $\mathsf{S}_{i,j,k} = 0$. Per $m = 6$ città europee appartenenti a Stati che si affacciano sul Mar Mediterraneo — Atene, Madrid, Nizza, Roma, Tirana e Zagabria — ricaviamo, per ogni anno, $max(\mathsf{SC}_{i,j})$, ovvero il valore record di giorni consecutivi di siccità. La Figura 2.1 mostra i valori della statistica appena creata nel tempo. Spiccano i dati di Madrid nel 2019, con 122 giorni consecutivi di siccità (dal 20 aprile al 25 agosto!), di Atene nel 2011 (97 giorni, dal 16 giugno al 20 settembre) e di Nizza nel 2017 (65 giorni, dal 29 giugno al 1° settembre). Non emergono però particolari trend ascendenti o discendenti, nemmeno modificando il valore soglia della variabile dicotomica $\mathsf{S}_{i,j,k}$.

Ci chiediamo allora se, più che la *durata* dei periodi siccitosi, sia aumentata la *frequenza* con cui questi si manifestano. In maniera piuttosto arbitraria, ma coerente con quanto osservato in Figura 2.1, stabiliamo come "siccitoso" un periodo senza precipitazioni superiori al millimetro giornaliero avente durata pari o superiore ai 20 giorni, e realizziamo la heatmap visibile in Tabella 2.2, nella quale tonalità tendenti al rosso indicano un anno con molti periodi siccitosi, e tonalità tendenti al verde un anno ricco di precipitazioni. Per le città di Nizza, Roma, Tirana e Zagabria, il numero annuo di periodi per i quali non ha piovuto per 20 o più giorni è aumentato soprattutto a partire dal 2017 in avanti. Madrid e Atene, invece, non hanno mai fatto registrare un anno con meno di due periodi siccitosi nel periodo di riferimento; da segnalare come la capitale della Grecia, nel 2022, abbia sperimentato assenza di piogge significative per lunghi lassi di tempo in ben 7 occasioni.

Sempre con riferimento all'analisi dei periodi siccitosi, spostiamo ora l'attenzione su alcune città del Nord Italia: Alessandria, Asti e Torino in Piemonte; Bergamo e Mantova in Lombardia; Belluno e Padova in Veneto; Modena in Emilia-Romagna. Come risaputo, soprattutto in quest'ultimo decennio l'Italia Settentrionale ha dovuto fare i conti sia con estati molto calde e afose, sia con inverni poco forieri di precipitazioni, e ciò ha causato un notevole stress idrico. Considerando ancora la definizione di "giorno piovoso" data in precedenza a questo stesso paragrafo, per il

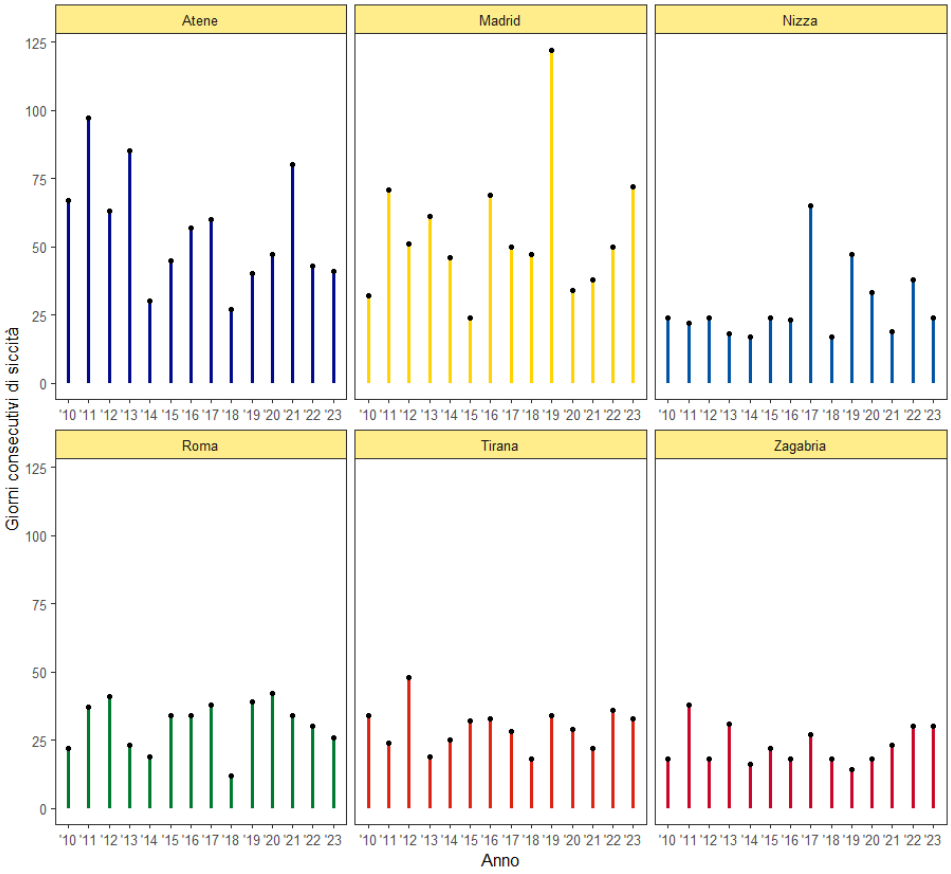


Figura 2.1: Record di giorni consecutivi di siccità in alcune città europee, anni 2010 – 2023.

periodo 2010-2023, deriviamo la statistica

$$\mathsf{SGT}_{i,j} = \sum_{k=1}^{365} \mathsf{S}_{i,j,k},$$

numero di giorni annui di siccità per la località i e l'anno j. Successivamente, prendendo spunto dal metodo di lavoro di Casadei & Finizio (2024), ricaviamo la deviazione del numero annuo di giorni di siccità dal dato del 2010, assunto come riferimento:

$$\mathsf{DEV\text{-}S}_{i,j} = \mathsf{SGT}_{i,j} - \mathsf{SGT}_{i,2010}.$$

L'andamento nel tempo della statistica ottenuta è nei grafici di Figura 2.2. Ovviamente, valori negativi di $\mathsf{DEV\text{-}S}_{i,j}$ corrispondono ad anni meno siccitosi rispetto al 2010. Quest'ultimo è stato un anno con frequenti perturbazioni: per certi versi, era

Tabella 2.2: Numero di periodi siccitosi in alcune città europee, anni 2010 – 2023.

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
Atene	5	3	4	3	6	5	3	4	4	3	4	3	7	3
Madrid	2	2	4	3	4	5	3	4	4	4	4	3	3	5
Nizza	1	1	1	0	0	2	2	3	0	3	3	0	2	4
Roma	1	2	1	1	0	2	2	2	0	3	1	2	3	4
Tirana	2	1	3	0	1	2	1	2	0	2	3	1	5	4
Zagabria	0	3	0	1	0	2	0	1	0	0	0	1	2	2

prevedibile ottenere conteggi dei giorni annui di siccità più alti rispetto al riferimento (testimoniati dalle linee con colorazione tendente al rosso nei grafici). Tuttavia, è impossibile non ravvisare una crescita molto marcata del valore di DEV-$S_{i,j}$ dal 2018 in poi, con una tendenza che raggiunge il picco nel 2022, *annus horribilis* dal punto di vista del meteo proprio sotto questo punto di vista. È evidente, inoltre, come la siccità abbia colpito particolarmente le città di Nord-Ovest, risparmiando leggermente quelle di Nord-Est, anche se perfino una città di montagna come Belluno ha visto aumentare il numero dei giorni totali di siccità nel corso del tempo — qui, nel 2022, sono stati registrati quasi 80 giorni di siccità in più rispetto al 2010.

Un'altra questione che sorge spontanea — preso atto dell'aumento, più o meno visibile, del numero annuale di giorni di siccità — riguarda l'impatto che hanno avuto i cambiamenti climatici sulle precipitazioni totali. È plausibile che i potenti temporali estivi, scaricando al suolo molta acqua in poco tempo, riescano in qualche modo a mascherare la mancanza di acqua, "falsificando" le statistiche annuali.

2.1.2 Precipitazioni piovose estive e invernali

Per le stesse città del Nord Italia considerate al Paragrafo 2.1.1 (Alessandria, Asti, Belluno, Bergamo, Mantova, Modena, Padova e Torino), andiamo a determinare quanto ha piovuto complessivamente nel periodo di riferimento, effettuando un confronto con la media storica (anziché con il primo dato disponibile, come fatto in Figura 2.2). A questo scopo, anzitutto, collezioniamo — sempre mediante il pacchetto R `openmeteo` — i dati pluviometrici dall'anno 1969 all'anno 2009 contenuti nella variabile precipitation_sum, e per ogni città ricaviamo, successivamente, la quantità di pioggia annuale media (in mm) dal 1969 al 2009. Seguendo la notazione presente in Wikle et al. (2019), par. 2.4.1, calcoliamo la somma delle precipitazioni

Figura 2.2: Differenze rispetto al 2010 dei giorni totali di siccità per alcune città dell'Italia settentrionale.

annue per la città s_i, indicata con $r_{z,s}(s_i)$, come

$$r_{z,s}(s_i) = \sum_{j=1}^{T} Z(s_i; t_j) \tag{2.1}$$

con $Z(s_i; t_j)$ millimetri di pioggia caduti nella città $s_i, i = 1, \ldots, m$, nel giorno $t_j, j = 1, \ldots, T$. Visto che le somme sono annuali e riferite a otto località, $m = 8$ e $T = 365$. Il vettore $\boldsymbol{r}_{z,s}$ racchiude l'ammontare precipitativo annuo per tutte le località:

$$\boldsymbol{r}_{z,s} = \begin{bmatrix} r_{z,s}(s_1) \\ \vdots \\ r_{z,s}(s_8) \end{bmatrix} = \left[\sum_{j=1}^{T} Z(s_1; t_j), \ldots, \sum_{j=1}^{T} Z(s_8; t_j) \right]^T. \tag{2.2}$$

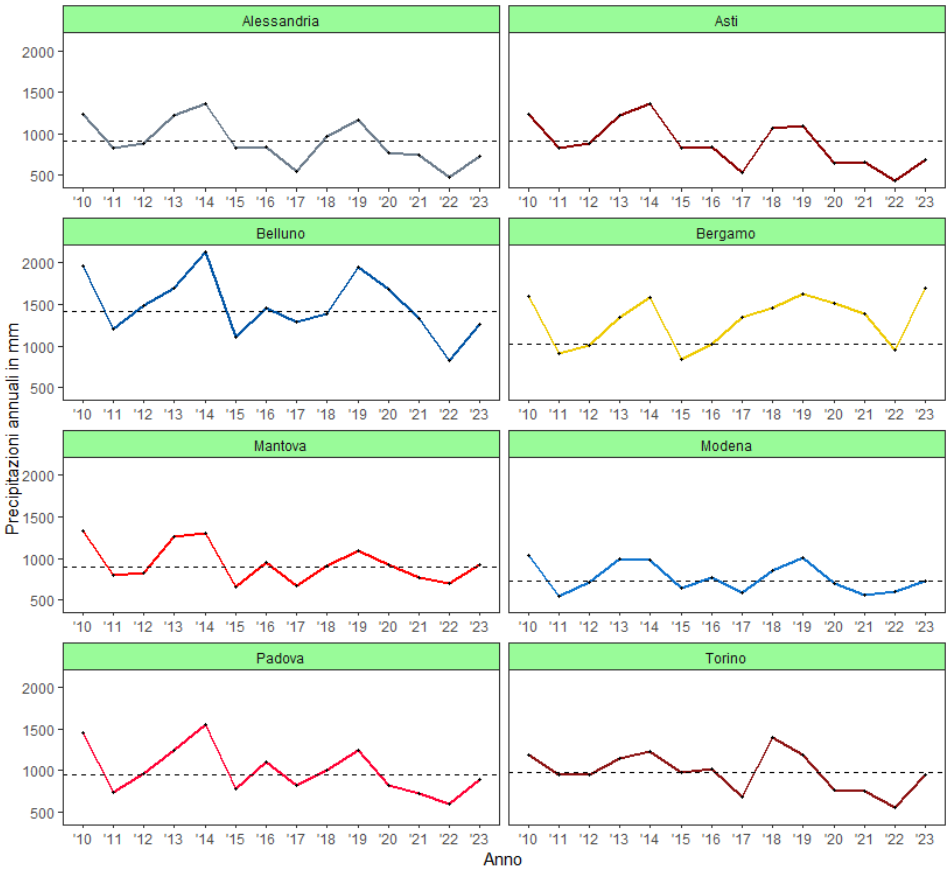


Figura 2.3: Serie storiche annuali delle precipitazioni in mm per alcune città dell'Italia settentrionale, nel periodo 2010 – 2023, e confronto con la media degli anni 1969 – 2009.

L'andamento di $\boldsymbol{r}_{z,s}$ nel corso del tempo è illustrato in Figura 2.3. Eccezion fatta per la serie storica relativa a Bergamo, che risulta essere quasi costantemente sopra media e ha un comportamento abbastanza peculiare, è agevole constatare come, specialmente per le città del Nord-Ovest, a partire soprattutto dal 2019 la carenza di precipitazioni si sia fatta sentire; di rilievo è il dato del 2022, che — come già evidenziato — è stato un anno particolarmente avaro di piogge.

Tutto sommato, però, non ravvisiamo allontanamenti (per così dire) "drammatici" dalla media storica. Filtriamo allora i dati per le otto città sopracitate prendendo in considerazione solo i mesi invernali (dicembre, gennaio e febbraio), e costruiamo grafici analoghi a quelli di Figura 2.3, raggruppandoli in Figura 2.4. In questo caso, la crisi idrica risulta essere ben più evidente, confermando il sospetto avanzato al termine del Paragrafo 2.1.1. In effetti, il problema della mancanza di

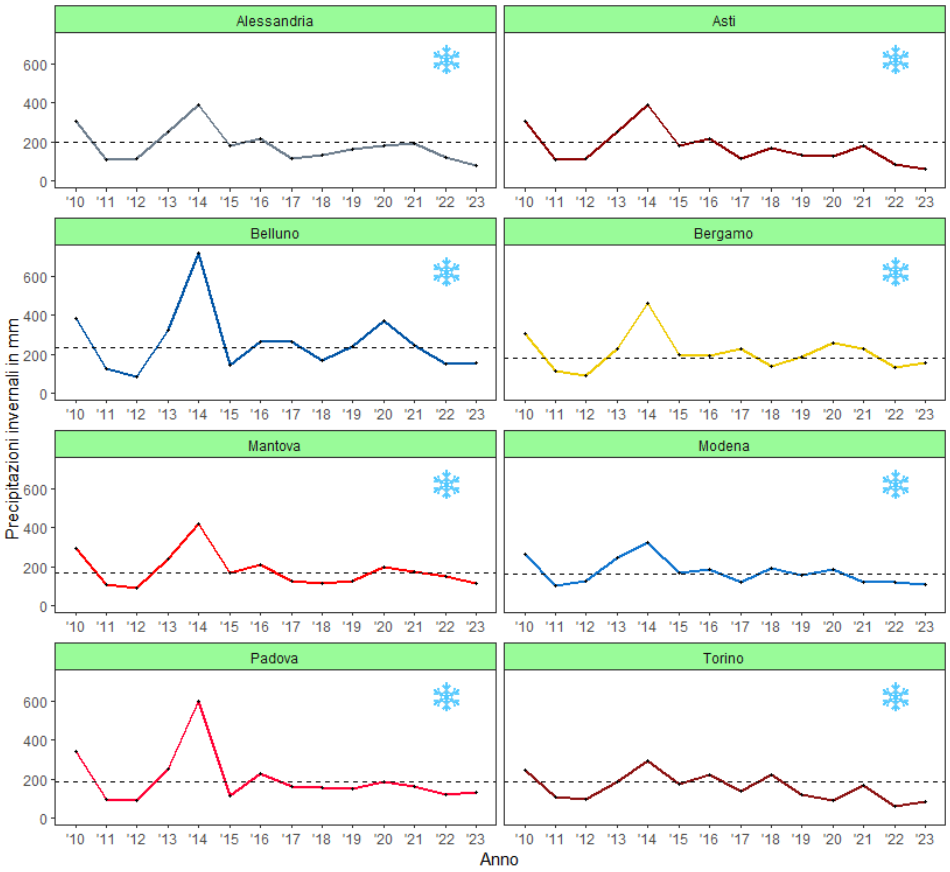


Figura 2.4: Serie storiche annuali delle precipitazioni invernali in mm per alcune città dell'Italia settentrionale, nel periodo 2010 – 2023, e confronto con la media degli anni 1969 – 2009.

pioggia non si limita temporalmente alla stagione estiva: durante l'inverno, l'anticiclone subtropicale africano sta iniziando a condizionare il tempo di gran parte dell'Europa occidentale, con le sue masse d'aria calda a bloccare affondi perturbati. La situazione più grave si osserva ancora per le città del Nord-Ovest, ma Padova è costantemente sotto la media storica dal 2017 e anche Mantova e Modena presentano numerosi valori inferiori alle precipitazioni attese.

2.1.3 Evoluzione delle temperature

Uno degli aspetti più preoccupanti del cambiamento climatico è sicuramente l'innalzamento della temperatura globale. Senza entrare nel merito delle nefaste problematiche connesse a questo fenomeno, ci limitiamo a rappresentare l'andamento

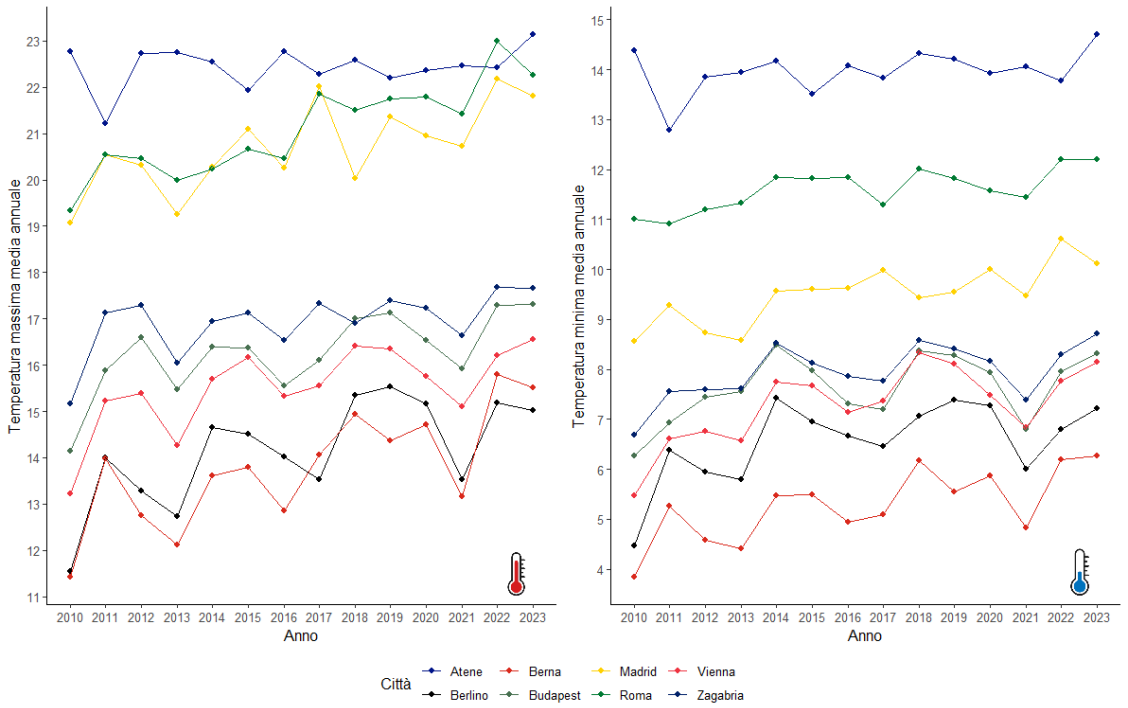


Figura 2.5: Serie storiche annuali della temperatura media massima (a sinistra) e minima (a destra) per alcune città europee, periodo 2010 – 2023.

della media annuale delle temperature massime e minime registrate per le città di Atene, Berlino, Berna, Budapest, Madrid, Roma, Vienna e Zagabria, visibile in Figura 2.5 per il consueto periodo di riferimento. Formalmente, detta $\hat{\mu}_{z,s}(s_i)$ la temperatura media annua (minima o massima) per la città s_i, calcoliamo stavolta (Wikle et al., 2019, par. 2.4.1)

$$\hat{\mu}_{z,s}(s_i) = \frac{1}{T}\sum_{j=1}^{T} Z(s_i; t_j), \tag{2.3}$$

con $Z(s_i; t_j)$ temperatura per la città $s_i, i = 1, \ldots, m$ nel giorno $t_j, j = 1, \ldots, T$. In questo caso, $m = 8$ e $T = 365$. Il vettore delle medie empiriche annue per le otto località diventa, allora,

$$\hat{\boldsymbol{\mu}}_{z,s} = \begin{bmatrix} \hat{\mu}_{z,s}(s_1) \\ \vdots \\ \hat{\mu}_{z,s}(s_8) \end{bmatrix} = \left[\frac{1}{T}\sum_{j=1}^{T} Z(s_1; t_j), \ldots, \frac{1}{T}\sum_{j=1}^{T} Z(s_8; t_j) \right]^T. \tag{2.4}$$

Abbiamo scelto appositamente città sparse per il continente europeo per vedere

se l'andamento è uniforme o presenta segnali di differenze geografiche. Le spezzate non lasciano adito a dubbi: sia per le massime, sia per le minime osserviamo trend crescenti per tutte le città considerate, segno che l'aumento delle temperature è un fenomeno veramente diffuso su scala planetaria e che recentemente ha avuto un'accelerazione considerevole. Ad esempio, Madrid è passata dai 19.1 °C di temperatura media massima annuale del 2010 ai 21.8 °C del 2023 (un incremento di 2.7 °C). A Vienna, invece, la temperatura media minima annuale del 2023 è stata di 8.1 °C, di ben 2.6 °C più alta rispetto a quella del 2010.

2.1.4 Trend delle nevicate

Altra tematica "scottante" nel contesto dei cambiamenti climatici è quella del diminuire della copertura nevosa sulle principali catene montuose (in particolare, Alpi e Pirenei). Inverni sempre più corti e caldi, nonché poveri — come visto al Paragrafo 2.1.2, di precipitazioni — hanno come logica conseguenza una minore quantità di neve disponibile per alimentare ghiacciai e riserve idriche. Ciò porta, per riflesso, ad una minore portata dei fiumi durante primavera ed estate. In Figura 2.6 mostriamo, rispetto alla media storica del periodo 1969 – 2009, l'andamento delle precipitazioni nevose in cm per otto città dove dovrebbe nevicare di frequente: Belluno, Biella, Cuneo e L'Aquila in Italia, Berna e Zurigo in Svizzera, Innsbruck in Austria e Grenoble in Francia. Da un punto di vista matematico, le equazioni utilizzate per ricavare l'informazione sono esattamente uguali alla (2.1) e alla (2.2), solo riferite alla variabile snowfall_sum.

Fatta eccezione per Grenoble, dove le somme annuali dei centimetri di neve caduta viaggiano sempre intorno alla media storica, dai grafici emerge ampiamente l'indicazione di una riduzione dei fenomeni nevosi già a partire dai primi anni del periodo di riferimento. Tale riduzione è particolarmente evidente per la città austriaca di Innsbruck. Infine, ulteriori approfondimenti sulla stessa tematica (qui non riportati) hanno rivelato come la quantità di neve caduta negli ultimi anni non sia diminuita solamente in montagna, ma anche — e più drasticamente — nelle città di pianura dove invece, stando ai riferimenti storici, nel passato la neve era ben più frequente.

2.2 Analisi dei dati orari

Anche per i dati a granularità oraria, come primo lavoro di analisi esplorativa presentiamo in Tabella 2.3 un compendio delle principali statistiche meteorologiche record del 2023 per i comuni della regione Veneto considerati. La temperatura massima più alta, 38.1 °C, è stata toccata ad Albaredo d'Adige (VR) nel tardo pomeriggio del 25 agosto, nel periodo di massima espansione di un promontorio

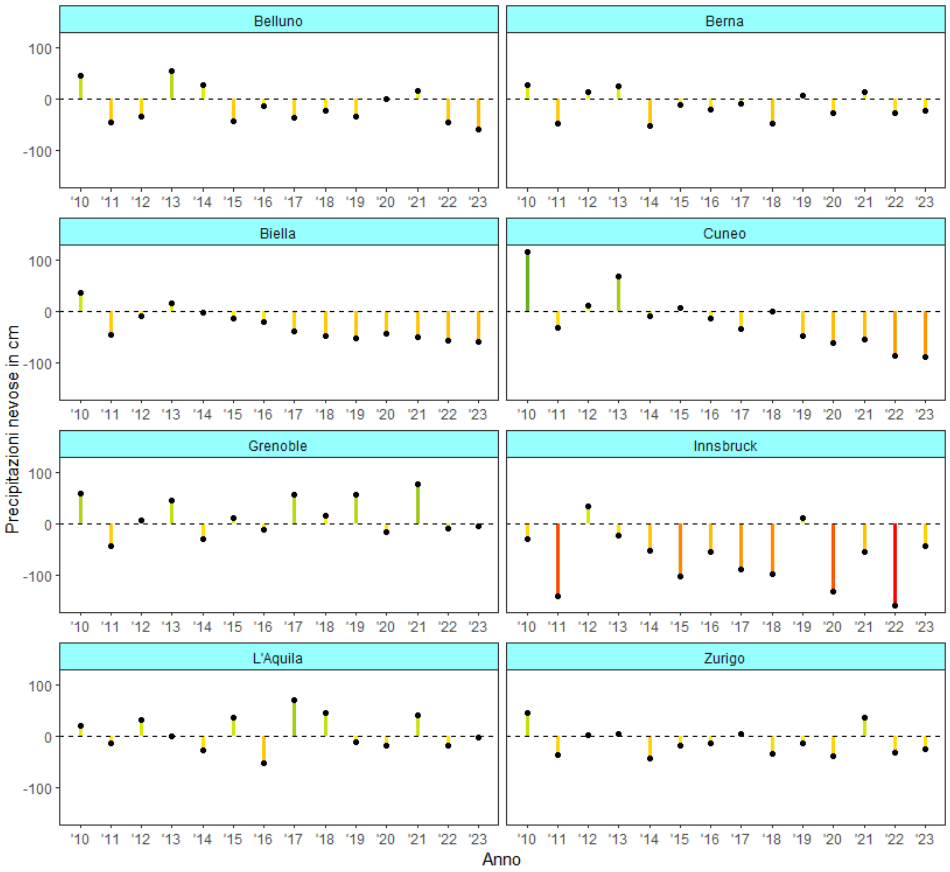


Figura 2.6: Precipitazioni nevose in cm di alcune città europee, anni 2010 – 2023, e confronto con la media degli anni 1969 – 2009.

di alta pressione. È a Cortina d'Ampezzo (BL) che si sono registrate, invece, la minima più bassa (−17.9 °C la mattina del 9 febbraio) e la raffica di vento più forte (114.5 km/h il 4 febbraio).

Dal dataset emerge inoltre come il record di precipitazione oraria, che appartiene a Bibione (VE), sia di 26.1 mm. Sebbene sia stato rilevato in un giorno in cui effettivamente vi sono state forti precipitazioni (il 25 luglio), purtroppo il dato appare ampiamente sottostimato, dal momento che, stando a quanto riportato nel Commento meteoclimatico ARPAV dell'estate 2023, i massimi di precipitazione oraria per il 2023 sono arrivati ad essere quasi il triplo. Una possibile causa di questa sottostima può risiedere nel fatto che i dati orari fanno riferimento a fasce temporali prefissate con inizio al primo minuto di ogni ora. Così, se un temporale inizia alle 9:30 e finisce alle 10:30, la quantità in mm di pioggia che ne deriva — pur essendo concentrata in un'ora — si ripartisce tra due fasce orarie diverse (quella

Tabella 2.3: Statistiche climatologiche sul dataset con i dati orari.

Record	Valore	Località	Data e ora
Temperatura massima	38.1 °C	Albaredo d'Adige (VR)	25/08, 17 – 18
Temperatura minima	−17.9 °C	Cortina d'Ampezzo (BL)	09/02, 05 – 06
Raffica di vento	114.5 km/h	Cortina d'Ampezzo (BL)	04/02, 09 – 10
Precipitazione oraria	26.1 mm	Bibione (VE)	25/07, 10 – 11
PM10	168.4 μg/m^3	Verona (VR)	16/02, 22 – 23
PM2.5	164.4 μg/m^3	Oderzo (TV)	22/06, 13 – 14
Monossido di carbonio	1386 μg/m^3	Verona (VR)	19/02, 20 – 21
Biossido di azoto	79.1 μg/m^3	Mirano (VE)	14/02, 23 – 00
Anidride solforosa	11.2 μg/m^3	Venezia (VE)	13/01, 19 – 20
Ozono	175 μg/m^3	Arzignano (VI)	25/08, 16 – 17
Ammoniaca	74.6 μg/m^3	Albaredo d'Adige (VR)	18/03, 07 – 08

con inizio alle 9:00 e quella con inizio alle 10:00). Ad un esame più accurato del dataset, infatti, scopriamo che a Bibione, il 25 luglio, sono caduti 20.3 mm di pioggia nella fascia oraria tra le 09:00 e le 10:00; per un totale di 46.4 mm da quello che probabilmente è stato un singolo temporale, dato più coerente con le rilevazioni ARPAV.

Le altre statistiche di Tabella 2.3 consistono nei picchi massimi della concentrazione degli inquinanti disponibili. La parte centrale del mese di febbraio 2023 è stata particolarmente inquinata in pianura, a causa di un regime meteo "di blocco" instaurato da un anticiclone; in provincia di Vicenza, ad Arzignano, il 25 agosto l'ozono presente nell'aria ha raggiunto i 175 μg/m^3, concentrazione di 75 μg/m^3 superiore al limite giornaliero consentito dall'Organizzazione Mondiale della Sanità.

2.2.1 Qualità dell'aria

Come già sottolineato esaminando la Tabella 2.3, la seconda decade del mese di febbraio 2023 per il Veneto è stata caratterizzata da un blocco anticiclonico che ha causato uno scarso rimescolamento dell'aria, e quindi un aumento delle concentrazioni di sostanze inquinanti. La mappa di Figura 2.7 rivela i livelli di PM10 nella fascia oraria dalle 18:00 alle 19:00 di giovedì 16 febbraio 2023, assieme ai nomi delle città con una maggiore e una minore presenza dell'inquinante (Verona e Cortina d'Ampezzo, rispettivamente). La grandezza dei puntini è proporzionale alla concentrazione di PM10, mentre colori dal giallo al nero indicano livelli pericolosi per la salute. Il valore limite giornaliero per questo tipo di particolato è infatti fissato dal D.Lgs. 155/2010 a 50 μg/m^3, e solamente 12 comuni su 54 — tutti situati nella parte nord-orientale della regione — erano sotto il limite. La situazione peggiore

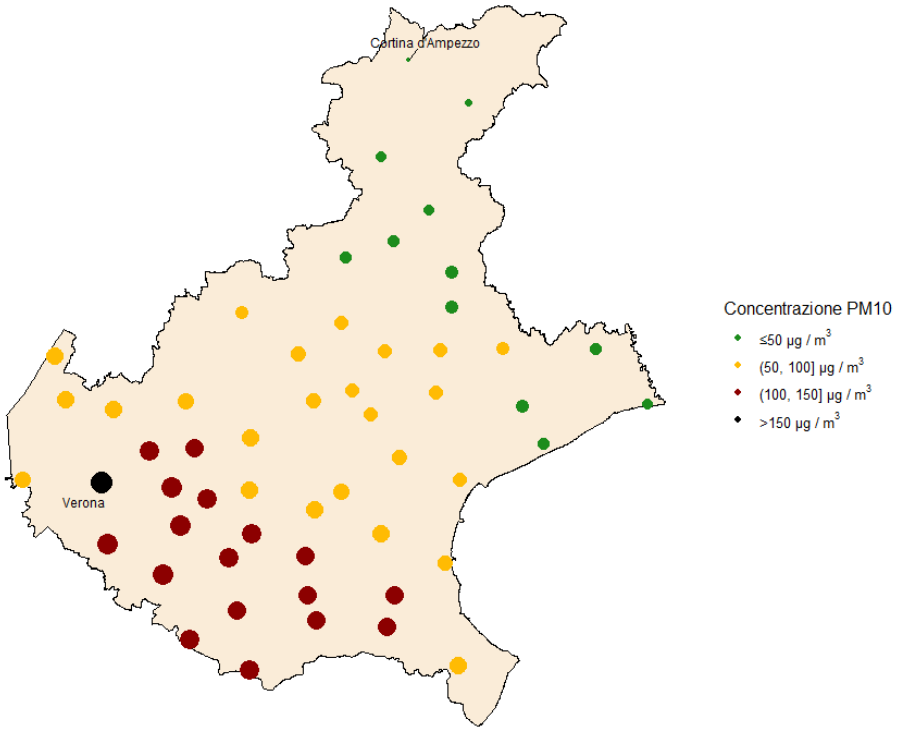


Figura 2.7: Concentrazione di PM10 per i 54 comuni veneti del dataset nella fascia oraria tra le 18:00 e le 19:00 del 16/02/2023.

si verificava per le località situate a sud-ovest, tra la Bassa Padovana e le province di Rovigo e Verona. Da questo grafico, con riferimento alla variabile pm10, sembra emergere la presenza di tre cluster di comuni ripartiti in tre aree geografiche (nord-orientale, centrale e sud-occidentale).

Per capire com'è stata complessivamente la qualità dell'aria nel corso dell'anno, è possibile far riferimento allo European AQI (Air Quality Index), indice costituito da un numero intero positivo. Un basso AQI è segno di un'aria pulita, mentre a valori alti dell'AQI corrisponde un'aria malsana. L'indice è calcolato a partire dalle concentrazioni di particolato (PM), diossido di azoto (NO_2), anidride solforosa (SO_2) e ozono (O_3) presenti nell'aria (Copernicus Atmosphere Monitoring Service, 2021). Per l'i-esimo comune veneto, $i = 1, \ldots, 54$, calcoliamo, a partire da tutti i dati disponibili,

$$\text{DEV-A}_i = \overline{\text{AQI}}_i - \overline{\overline{\text{AQI}}},$$

deviazione del suo AQI medio rispetto alla media regionale. Come vediamo dalla

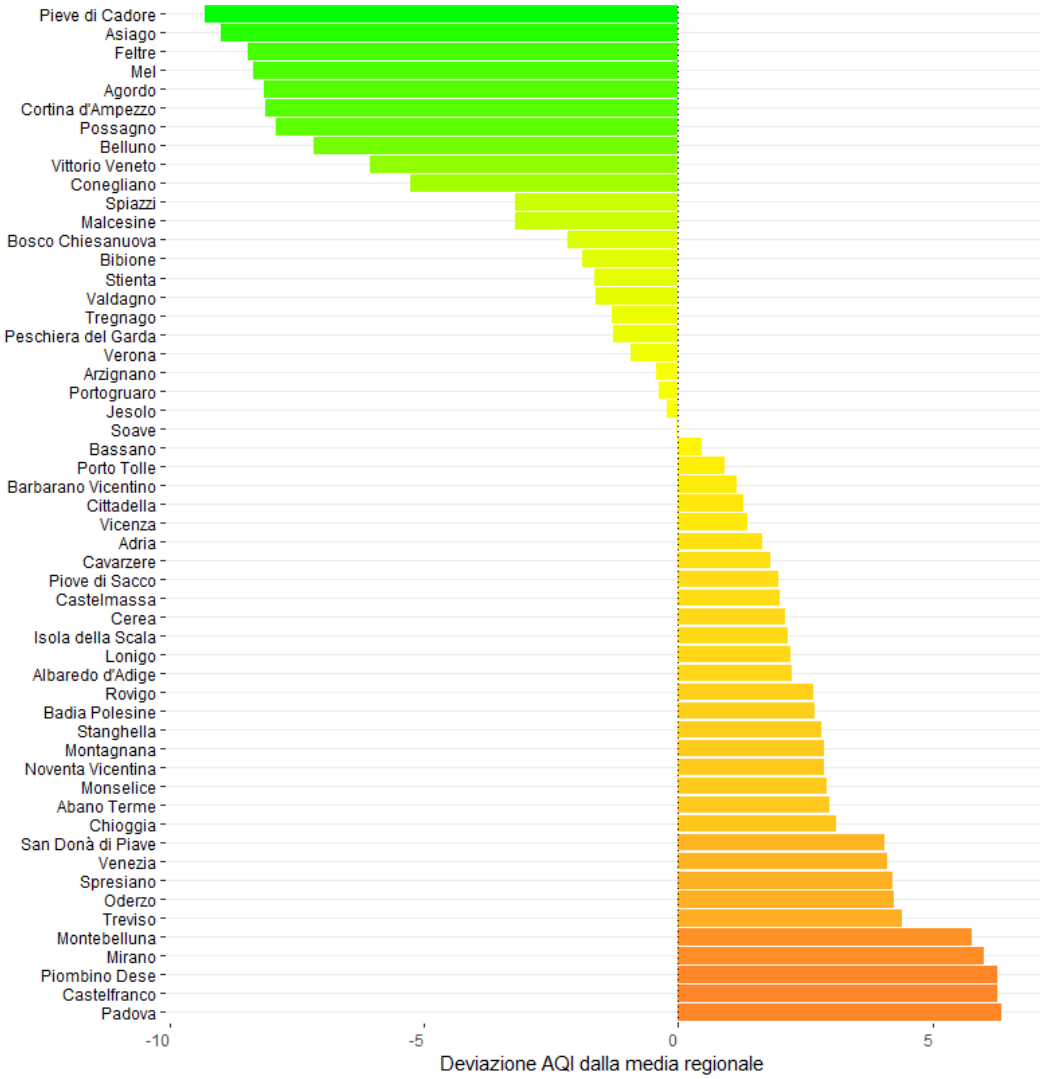


Figura 2.8: Deviazioni dall'AQI (Air Quality Index) medio per i 54 comuni veneti del dataset.

Figura 2.8, le località situate in zone montuose e quelle sul Lago di Garda hanno un AQI inferiore alla media regionale. C'è una discreta variabilità, invece, per quelle vicine al mare: alcune, come Bibione (VE), Jesolo (VE) e Porto Tolle (RO) hanno un indice inferiore alla media o solo leggermente sopra la media, mentre Chioggia (VE) e soprattutto Venezia sono ben sopra la media. Gli indici delle città di pianura, in particolare quelle del padovano e del veronese, sono infine quasi tutti sopra la media.

Tabella 2.4: Classificazione dell'instabilità atmosferica in base al CAPE secondo il National Weather Service (NWS).

Instabilità atmosferica	CAPE (J/kg)
Debole	<1000
Moderata	1000-2500
Forte	2500-4000
Estrema	>4000

2.2.2 L'indice CAPE

Il CAPE, acronimo di convective available potential energy, è un indice che esprime l'ammontare di energia potenziale disponibile per la formazione di nubi convettive (come i cumulonembi). Misurato generalmente mediante strumentazioni agganciate a palloni aerostatici, il CAPE rappresenta una sorta di livello di instabilità dell'atmosfera, e per questo motivo è considerato un indice molto prezioso per le previsioni di fenomeni temporaleschi intensi. La classificazione dell'instabilità atmosferica in base al CAPE proposta dal National Weather Service, l'agenzia governativa statunitense che si occupa delle previsioni meterologiche — ovviamente applicabile anche con riferimento al territorio europeo — è visibile in Tabella 2.4.

Il pacchetto R openmeteo non permette di estrarre direttamente i valori del CAPE; tuttavia, questi sono disponibili in formato GRIB nel sito web di ERA5. Una volta effettuato il download, specificando le coordinate geografiche e il periodo di interesse, abbiamo convertito i dati con il codice Python visibile in Appendice nello stesso formato degli altri dati orari, e infine abbiamo unificato i due dataset.

Nella Figura 2.9 mostriamo i valori dell'indice CAPE registrati nel Veneto centro-meridionale tra le 18:00 e le 19:00 del 19 luglio 2023. La terza decade del mese di luglio 2023, sia per il Veneto, sia per la maggior parte dell'Italia settentrionale, è stata caratterizzata da un tempo in prevalenza instabile a causa dell'azione del flusso umido e fresco atlantico, sceso di latitudine. Nella seconda decade, invece, un imponente anticiclone di matrice africana aveva determinato tempo soleggiato e stabile con temperature sopra media, contribuendo ad aumentare l'energia potenziale immagazzinata nell'atmosfera. È impressionante notare infatti come, specialmente nelle aree vicine al mare Adriatico e nel mare Adriatico stesso, siano stati rilevati picchi di CAPE superiori a 6000 J/kg, segno di un'instabilità atmosferica estrema secondo la Tabella 2.4: preludio dei violenti temporali con grandine e forti raffiche di vento che si sarebbero verificati nei giorni seguenti.

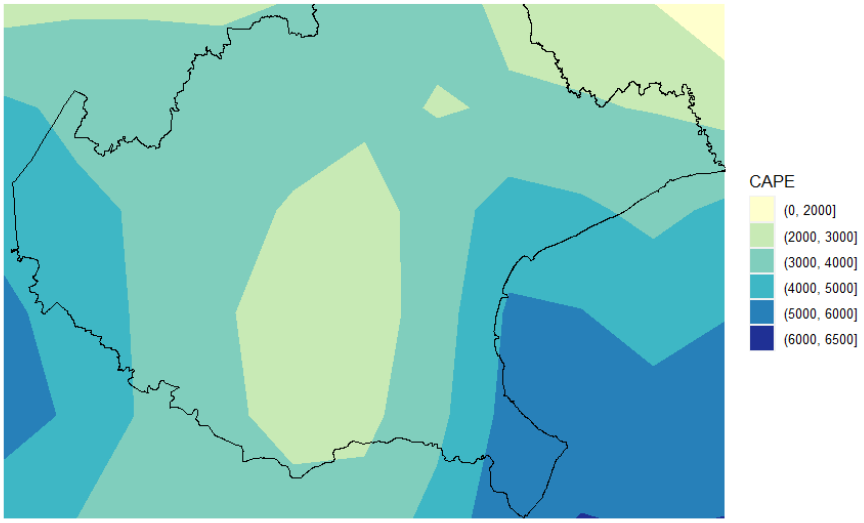


Figura 2.9: Valori dell'indice CAPE nel Veneto centro-meridionale alle ore 18:00 del 19 luglio 2023.

2.3 Tecniche di clustering

Sfruttando le variabili disponibili, possiamo andare alla ricerca di città con caratteristiche climatiche simili. Per far questo, lo strumento statistico d'elezione è il clustering, che rientra nelle tecniche di apprendimento non supervisionato. In particolare, un algoritmo di raggruppamento molto utilizzato è il cosiddetto k-means, che si prefigge di ottenere una partizione di n elementi in G gruppi, con $G < n$ fissato a priori. Le unità statistiche — nel contesto dei dati meteorologici in esame, le città — sono spostate da un gruppo all'altro fino ad ottenere una partizione che soddisfi il criterio della massima omogeneità intra-gruppo e massima eterogeneità tra gruppi; la procedura si ferma al raggiungimento di un certo criterio di arresto, ad esempio un livello minimo di diminuzione della devianza entro i gruppi, W (Bassi & Ingrassia, 2022).

2.3.1 Città europee e del Nordafrica

Applichiamo la tecnica k-means al dataset con granularità giornaliera, allo scopo di svelare le città con tratti climatici simili. Trasformiamo il dataset in formato largo, e ricaviamo per ogni mese di ogni anno dal 2010 al 2023 le medie delle variabili di raggruppamento. Queste ultime — standardizzate prima di far partire la procedura

— sono quelle relative a temperatura, vento, accumuli pluviometrici, irraggiamento ed evapotraspirazione. Il numero di gruppi G viene arbitrariamente posto pari a 4; un valore più grande di G porterebbe ovviamente a identificare zone climatiche più circoscritte.

In Figura 2.10 possiamo vedere i risultati della procedura, suddivisi per anno. A catturare l'attenzione è il punto corrispondente a Laghouat, al centro dell'Algeria: sorprendentemente, le sue caratteristiche climatiche risultano più simili alle città della Pianura Padana o addirittura dell'Europa orientale che non a quelle delle altre città nordafricane o spagnole. Guarda caso, la città algerina sorge ad un'altitudine di 769 m e, in base alla classificazione del clima di Köppen, ha un "clima freddo del deserto" (Finlayson et al., 2007). Le città sicule, greche, del Nordafrica e della Spagna meridionale vengono quasi sempre raggruppate nello stesso cluster, così come quelle del Nord Europa. Al contrario, le città dell'Italia centro-settentrionale e della Francia centro-meridionale non vengono assegnate sempre allo stesso cluster, segno di un clima particolarmente variabile nel corso del periodo di tempo considerato.

2.3.2 Città venete

L'obiettivo dichiarato all'inizio del paragrafo precedente può essere raggiunto anche per le città venete del dataset a granularità oraria. Stavolta, oltre che per le caratteristiche meteorologiche, possiamo raggruppare tenendo conto delle variabili sulla qualità dell'aria. Prima di tutto, imputiamo a 0 i dati mancanti relativi alle variabili alder_pollen, birch_pollen, grass_pollen, mugwort_pollen, olive_pollen, ragweed_pollen perché, ragionevolmente, se non è attivo il monitoraggio dei pollini di un certo tipo vuol dire che — per fattori stagionali — la concentrazione di pollini nell'aria è nulla. Poi, rendiamo il dataset in formato largo e per ogni città calcoliamo i valori medi di ciascuna variabile per ogni mese del 2023.

Dalla Figura 2.7 potevamo riconoscere la presenza di tre cluster di comuni. Per vedere se effettivamente è così, prendendo però in esame tutte le variabili disponibili (non solo pm10) e tutto il 2023, creiamo il grafico di Figura 2.11: in funzione del numero di cluster, mettiamo i valori della percentuale di devianza spiegata dai cluster, W, calcolata mediante il rapporto

$$W = \frac{B}{T},$$

dove B è la devianza tra i cluster e T è quella totale. Effettivamente, il "gomito" del grafico si ottiene approssimativamente in corrispondenza di 3 cluster; il miglioramento apportato dal considerarne 4 non è molto evidente. Mostriamo su mappa, in Figura 2.12, l'appartenenza di ogni comune ad un cluster. Uno di questi comprende principalmente comuni situati su Alpi e Prealpi, per i quali evidentemente il clima

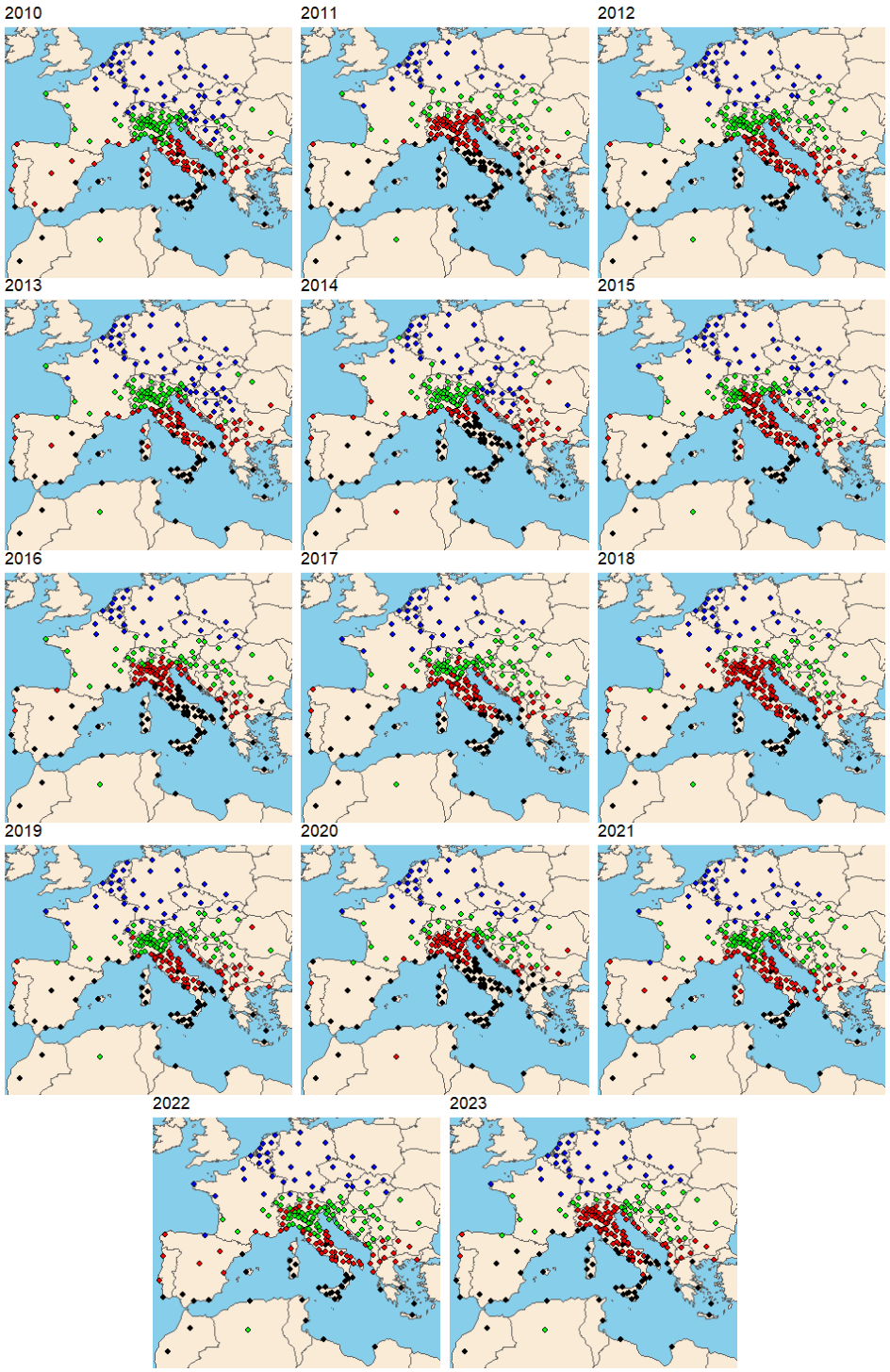


Figura 2.10: Clustering delle città del dataset a granularità giornaliera per anno.

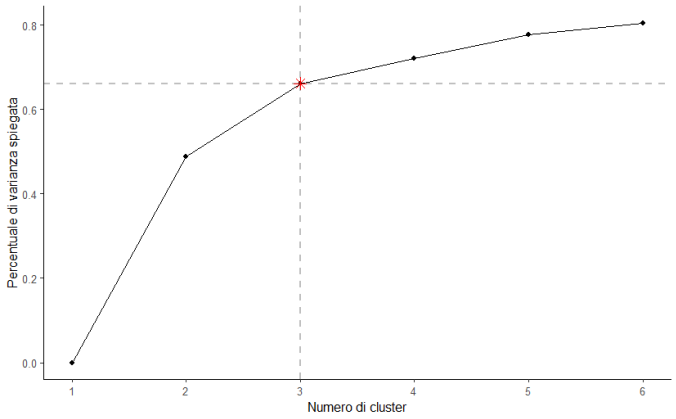


Figura 2.11: Individuazione del numero ottimale di cluster (dati a granularità oraria).

si mantiene non eccessivamente umido per tutto l'anno e l'aria è più pulita (come desumiamo, del resto, dalla Figura 2.8). Un secondo cluster raggruppa le località vicine alla costa o al Lago di Garda, dove i tassi di umidità forse sono più elevati rispetto ai comuni del primo cluster, ma che tendenzialmente beneficiano di venti più sostenuti rispetto alla media. Il terzo e più numeroso cluster raggruppa le città di pianura, caratterizzate da un clima nebbioso in inverno e afoso d'estate, con scarsa ventilazione se non in concomitanza di periodi perturbati — cosa che favorisce l'accumularsi di inquinanti nell'aria. Peculiari sono le collocazioni di Peschiera del Garda (VR), Belluno e Mel (BL): la prima risulta appartenere al cluster di pianura, mentre le città bellunesi sono nel cluster con quelle di mare e di lago.

2.3.3 Analisi dell'andamento della dissimilarità tra cluster

L'analisi dei cluster sviluppata per i dati sui comuni veneti, ma anche per le città europee e africane al Paragrafo 2.3.1, ha utilizzato per il raggruppamento quantità ottenute come media di variabili meteorologiche calcolate su un periodo di un mese. Tali medie rischiano di aver "neutralizzato" buona parte della variabilità insita in misurazioni ad elevata volatilità come quelle atmosferiche. Vogliamo scoprire di quanto cambierebbe la composizione dei gruppi se il clustering fosse fatto su base settimanale, anziché mensile, e come varierebbe nel tempo la dissimilarità tra cluster.

Prendiamo nuovamente in esame i dati orari sull'inquinamento. L'obiettivo, ora, è di visualizzare l'andamento settimanale di una misura della dissimilarità tra la qualità dell'aria rilevata in insiemi di comuni ad alta, media e bassa concentrazione di inquinanti. Il procedimento che adottiamo è il seguente.

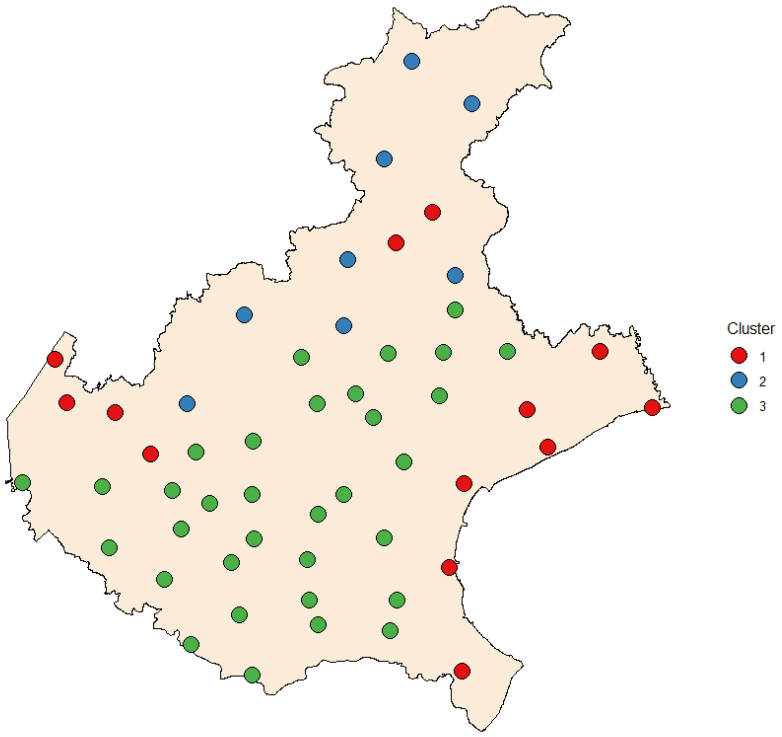


Figura 2.12: Appartenenza di ciascun comune veneto ai cluster definiti con k-means, considerando tutte le variabili.

1. Costruzione di un dataset in formato largo con le medie delle variabili orarie sull'inquinamento (pm_10, pm2_5, carbon_monoxide, nitrogen_dioxide, sulphur_dioxide, hourly_ozone, ammonia) per ogni settimana del 2023, per ogni città.

2. Standardizzazione delle variabili.

3. Per ognuna delle 52 settimane del 2023:

   - procedura di clustering con k-means impostando 3 cluster;
   - calcolo della matrice delle distanze euclidee tra i centroidi dei 3 cluster, D:

$$D = \begin{bmatrix} 0 & D_{1,2} & D_{1,3} \\ D_{2,1} & 0 & D_{2,3} \\ D_{3,1} & D_{3,2} & 0 \end{bmatrix};$$

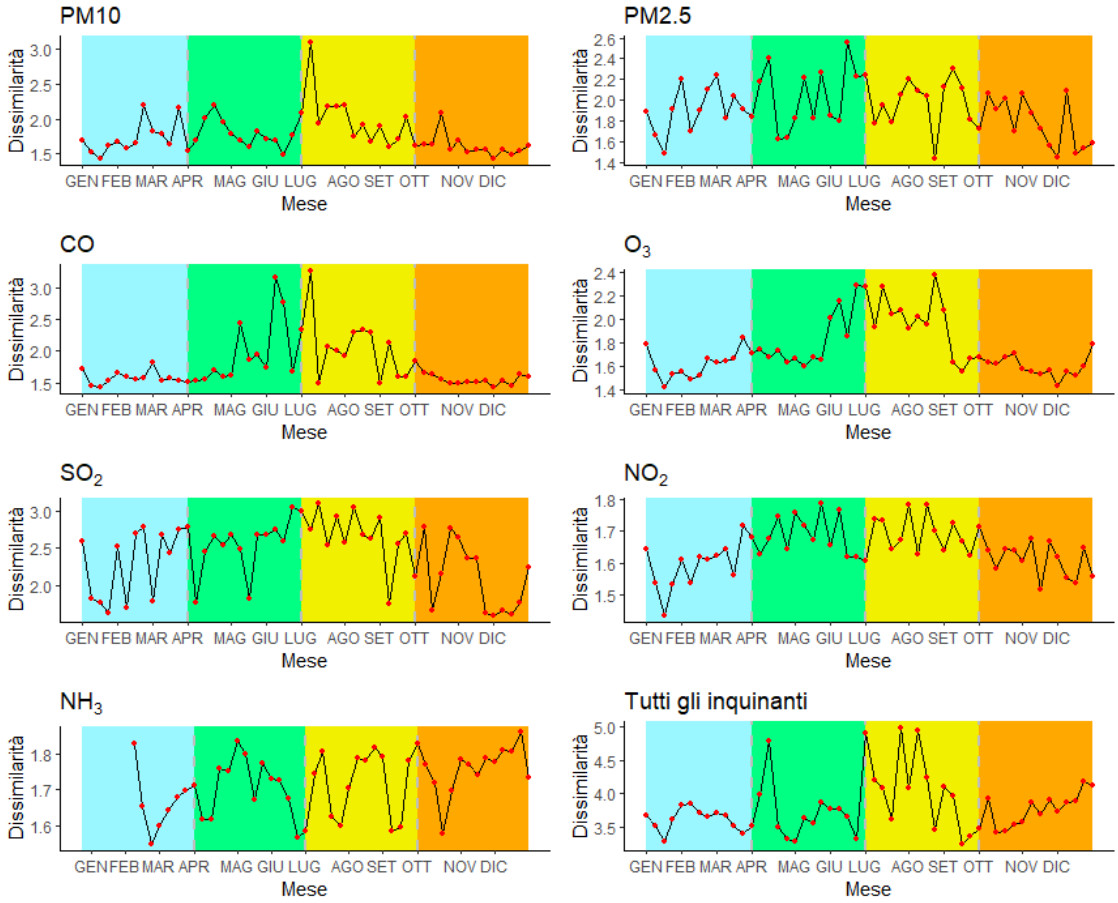


Figura 2.13: Andamento temporale dell'indice di dissimilarità tra cluster ricavati con k-means.

- calcolo della distanza media tra i centroidi dei cluster, $\overline{D} = \frac{D_{1,2}+D_{2,3}+D_{1,3}}{3}$, considerabile come una misura della loro dissimilarità: tanto minore $\overline{D}$, tanto più vicini i centroidi e tanto più simili le concentrazioni di inquinanti.

Riportiamo in Figura 2.13 l'andamento del tempo della misura di dissimilarità. Gli sfondi di ogni grafico sono arricchiti da un colore che simboleggia la stagione dell'anno (azzurro per l'inverno, verde per la primavera, giallo per l'estate e arancione per l'autunno). Per il monossido di carbonio (CO) apprezziamo come, a cavallo tra la primavera e l'estate, vi sia un picco della differenza tra le concentrazioni di questo inquinante nei tre cluster. Anche per l'ozono (O_3), il diossido di azoto (NO_2) e le PM10, ma specialmente per il primo composto, le dissimilarità tra cluster si palesano principalmente nei periodi primaverile ed estivo. Per gli altri inquinanti

— anidride solforosa (SO_2), ammoniaca (NH_3) e particolato fine (PM2.5) — l'andamento dell'indice di dissimilarità costruito è molto irregolare. Abbiamo realizzato il grafico in basso a destra in Figura 2.13 effettuando il clustering con tutte le variabili citate. Nell'anno 2023, le differenze nella concentrazione di inquinanti in Veneto tra gruppi di comuni sono state più consistenti tra aprile e maggio e tra luglio e settembre.

La procedura appena descritta, da sola, non permette di capire *quali* comuni siano finiti dentro a *quale* cluster per ognuna delle ripetizioni di k-means, ma solo di ottenere una misura della diversità di concentrazioni di inquinanti nel territorio nell'arco del 2023. L'ammoniaca viene generata principalmente dalle attività agricole e dagli allevamenti intensivi; quindi, è sensato constatare l'assenza di andamenti stagionali nella differenza di concentrazione di questo inquinante tra città di pianura, montagna e mare. Le altre sostanze derivano principalmente dalle emissioni veicolari e dalla combustione (negli impianti domestici e industriali), con l'eccezione dell'ozono, che si forma per reazione fotochimica grazie alla luce del Sole, ma necessita comunque della presenza in aria di altri inquinanti. Risulta però difficile, senza conoscenze aggiuntive — come la composizione dei singoli cluster in termini di comuni — motivare l'andamento degli altri grafici.

Per dare un'idea della composizione settimanale dei gruppi proponiamo la Figura 2.14, costituita da 12 mappe del Veneto relative alle prime settimane di ogni mese del 2023. Tali mappe vanno ad integrare l'informazione ottenuta con la misura di dissimilarità. Puntini di colore, rispettivamente, blu, arancione e grigio indicano l'appartenenza di un certo comune al cluster con bassi, intermedi ed alti valori di inquinanti. Il mese nel quale il cluster con alta concentrazione di inquinanti risulta essere popolato dal minor numero di comuni è giugno; discorso inverso per febbraio e settembre.

Figura 2.14: Composizione dei cluster per le prime settimane di ogni mese del 2023. Dati sull'inquinamento per i 54 comuni del Veneto.

Capitolo 3

Il modello hidden Markov

Una famiglia di modelli che consente di andare oltre l'assunzione di indipendenza e identica distribuzione delle osservazioni, particolarmente adatta ad essere utilizzata nel mondo della statistica applicata alla meteorologia (ma non solo), è quella dei modelli di Markov a stati nascosti (hidden Markov models, HMMs). Nel presente Capitolo forniremo una panoramica degli elementi statistici caratterizzanti gli HMMs. I principali riferimenti letterari sono Cappé et al. (2005) e Zucchini et al. (2016).

3.1 Gli hidden Markov models

Gli hidden Markov models sono una classe di modelli nei quali la distribuzione che genera un'osservazione y_i dipende dallo stato di un processo di Markov latente (cioè inosservabile). Proposti per la prima volta da Leonard E. Baum negli anni Sessanta (Baum & Petrie, 1966), questi modelli, che fanno perno sulla proprietà di Markov, sono molto versatili e si prestano a descrivere l'evoluzione sequenziale di un fenomeno.

3.1.1 Catene di Markov e misture

In questo paragrafo introduciamo i due principali concetti alla base di un modello hidden Markov: le catene di Markov e i modelli a mistura finita.

Catene di Markov

Una sequenza di variabili casuali discrete $X_0, X_1, \ldots, X_t$ è una catena di Markov (Markov Chain, abbreviata in MC) del prim'ordine se soddisfa la seguente relazione, per l'appunto detta "proprietà Markoviana":

$$\mathbb{P}(X_t|x_0, x_1, \ldots, x_{t-1}) = \mathbb{P}(X_t|\boldsymbol{x}^{(t-1)}) = \mathbb{P}(X_t|x_{t-1}) \quad \forall t = 1, 2, \ldots \tag{3.1}$$

Questo significa che la distribuzione di X_t condizionata ai valori assunti dalle variabili $X_0, X_1, \ldots, X_{t-1}$ è uguale alla distribuzione condizionata di X_t rispetto al solo ultimo valore assunto dal processo, x_{t-1}.

Consideriamo ora una catena del prim'ordine, e supponiamo che le possibili realizzazioni delle variabili casuali discrete X_t siano nell'insieme

$$\{1, 2, \ldots, m\},$$

cioè che lo spazio degli stati ("state space") di X_t sia finito e conti m stati. Una catena di Markov è omogenea se la probabilità di trovarsi in un certo stato j, al tempo $s+t$, condizionatamente al fatto di essersi trovati nello stato i, al tempo s, non è legata al particolare s. In altre parole, le probabilità di transizione condizionate

$$\gamma_{ij}(t) = \mathbb{P}(X_{s+t} = j|X_s = i), \quad i, j = 1, \ldots, m, \tag{3.2}$$

non dipendono dallo specifico s, ma solo dal ritardo (o "lag") t.

Sia poi $\boldsymbol{\Gamma}(t)$ la matrice con elementi $\gamma_{ij}(t)$, chiamata "matrice delle probabilità di transizione" e abbreviata in t.p.m. Le catene di Markov omogenee soddisfano la proprietà di Chapman-Kolmogorov secondo cui

$$\boldsymbol{\Gamma}(s+t) = \boldsymbol{\Gamma}(s)\boldsymbol{\Gamma}(t) \Rightarrow \boldsymbol{\Gamma}(t) = \boldsymbol{\Gamma}(1)^t = \boldsymbol{\Gamma}^t \quad \forall t \in \mathbb{N}, \tag{3.3}$$

dove con $\boldsymbol{\Gamma}(1) = \boldsymbol{\Gamma}$ denotiamo la matrice (quadrata) delle probabilità di transizione tra gli m stati latenti a ritardo 1. Naturalmente, la somma in j degli elementi γ_{ij} è pari a 1: tale è la probabilità di transitare in uno a caso degli altri $m-1$ stati a partire da un certo stato i, o di rimanere in i. Questo importante risultato permette di calcolare la t.p.m. ad un qualsiasi istante temporale in maniera molto semplice, e verrà sfruttato nelle applicazioni della Sezione 3.2, per costruire delle previsioni. Nel seguito, gli indici i e j faranno riferimento, rispettivamente, alle righe e alle colonne della t.p.m, che quindi sommerà a 1 per riga.

Una catena di Markov con matrice delle probabilità di transizione $\boldsymbol{\Gamma}$ ha una distribuzione stazionaria $\boldsymbol{\delta} = (\delta_1, \delta_2, \ldots, \delta_m)$, $\delta_i \geq 0 \ \forall i = 1, \ldots, m$, se — sotto condizioni di ergodicità e aperiodicità — si verifica:

— $\boldsymbol{\delta}\boldsymbol{\Gamma} = \boldsymbol{\delta}$ (stazionarietà);

— $\boldsymbol{\delta}\mathbf{1}' = 1$ (affinché $\boldsymbol{\delta}$ sia effettivamente una distribuzione di probabilità).

La distribuzione stazionaria, quindi, non cambia se post-moltiplicata per $\boldsymbol{\Gamma}$. Per catene di Markov omogenee e definite su uno spazio finito di stati, si può dimostrare che la distribuzione stazionaria viene raggiunta per $t \to \infty$.

Modelli a mistura finita

Una variabile casuale Y segue una distribuzione a mistura finita se la sua funzione di probabilità è data da una combinazione di distribuzioni di probabilità, con pesi definiti da una variabile di mistura ("mixing variable") X:

$$\begin{aligned} p(y) &= \sum_{i=1}^{m} \delta_i p_i(y) \\ &= \sum_{i=1}^{m} \mathbb{P}(X = i) p(y|i), \end{aligned} \tag{3.4}$$

dove

$$X = \begin{cases} 1 & \text{con probabilità } \delta_1 \\ 2 & \text{con probabilità } \delta_2 \\ \dots & \\ m & \text{con probabilità } \delta_m \end{cases}. \tag{3.5}$$

Le realizzazioni di X sono considerate indipendenti l'una dall'altra (a differenza di quanto accade per gli HMMs, come verrà evidenziato in seguito); valgono relazioni analoghe a quelle presentate nel caso continuo. I modelli a mistura finita trovano impiego anche nell'ambito del marketing, ed in particolare nel clustering model-based (Grün, 2019).

3.1.2 Specificazione del modello

Ogni modello di Markov a stati latenti si compone di:

1. una catena di Markov a tempo discreto $(X_t)_{t\geq 1}$ latente (cioè non osservabile), dotata delle proprietà viste al Paragrafo 3.1.1;

2. un processo a tempo discreto $(y_t)_{t\geq 1}$ osservabile.

Data una realizzazione x_t della catena di Markov, le y_t sono condizionatamente indipendenti; ovvero, X_t (e solo X_t) governa la distribuzione delle Y_t. Per la proprietà

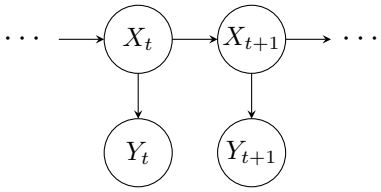


Figura 3.1: Rappresentazione grafica della struttura di dipendenza presente in un modello hidden Markov (Cappé et al., 2005).

Markoviana (3.1), inoltre, X_{t+1} dipende solamente da X_t, lo stato precedente della catena. La struttura del modello, visibile graficamente in Figura 3.1, è

$$\begin{cases} \mathbb{P}(X_t|\boldsymbol{X}^{(t-1)}) = \mathbb{P}(X_t|X_{t-1}), & t = 1, 2, \dots \\ p(y_t|\boldsymbol{y}^{(t-1)}, \boldsymbol{x}^{(t)}) = p(y_t|x_t), & t \in \mathbb{N}. \end{cases} \tag{3.6}$$

La prima equazione della (3.6) è detta "di transizione" e rappresenta l'evoluzione dinamica degli stati latenti dettata dalla catena di Markov. Il fatto che non si possano osservare direttamente le realizzazioni degli stati latenti giustifica l'aggettivo "hidden" presente nel nome di questi modelli. La seconda equazione è quella "di misura" e fa riferimento alle variabili osservate a cui, per via della loro dipendenza dagli stati latenti, è prassi far riferimento con l'appellativo di "variabili state-dependent".

Il modello hidden Markov può essere visto come un caso particolare dei modelli a mistura finita introdotti al Paragrafo 3.1.1. Infatti, la distribuzione marginale di Y_t, ricavabile sfruttando la legge delle probabilità totali, è analoga, nella forma, a quella vista nell'Equazione 3.4. L'unica differenza tra HMMs e modelli a mistura finita, come anticipato, risiede nella struttura di dipendenza che sussiste tra le X_t: gli hidden Markov models sono misture finite con dipendenza di tipo Markoviano tra le variabili di stato.

$$p(Y_t = y) = \sum_{i=1}^{m} \underbrace{p(Y_t = y|X_t = i)}_{g(y_t|i)} \underbrace{\mathbb{P}(X_t = i)}_{\text{mixing weights}} \tag{3.7}$$

Analogie con i modelli state-space

Fino a questo momento, la variabile di stato X_t che segue un processo di Markov è stata considerata come discreta. In realtà, in molte applicazioni l'uso di un certo numero m di stati assumibili da X non trova giustificazioni empiriche. Pensiamo, ad esempio, alla modellazione del numero annuale di giorni con vento superiore a 100

km/h in una certa località geografica con un HMM a 3 stati latenti. Da un punto di vista prettamente meteorologico, non esistono ragioni precise per considerare 3 stati latenti piuttosto che 4, 5 o un qualsiasi altro numero finito.

I modelli state-space (SSMs) condividono in toto la struttura degli HMMs, ma considerano la variabile di stato X_t come continua. In realtà, è opportuno sottolineare come, in letteratura, alcuni autori — ad esempio Cappé et al. (2005) — non facciano una vera e propria distinzione tra modelli state-space e hidden Markov models, bensì cataloghino questi ultimi come casi particolari dei primi aventi uno state space discreto. La struttura di un generale SSM è la seguente:

$$\begin{cases} \alpha_{t+1} = T_t \alpha_t + \eta_t & t = 1, 2, \ldots, T+1 \\ y_t = Z_t^T \alpha_t + \epsilon_t & t = 1, 2, \ldots, T \\ \alpha_1 \sim \mathsf{N}(\alpha_{1|0}, P_{1|0}) \end{cases}, \tag{3.8}$$

dove le variabili latenti sono state indicate con α_t, T_t è la matrice di transizione tra stati e Z_t è una matrice di pesi. Come per la (3.6), le due equazioni della (3.8) sono chiamate, rispettivamente, equazione di transizione e di misura. I modelli state-space sono anche noti come "modelli di regressione con parametri time-varying" o "modelli di regressione dinamica", poiché permettono di estendere la regressione classica modellando l'evoluzione nel tempo dei parametri. Per di più, possono essere anche visti come particolari modelli misti nei quali gli effetti casuali α_t possiedono una struttura di dipendenza Markoviana.

3.1.3 Inferenza negli hidden Markov models

Oltre alla stima dei parametri, gli obiettivi inferenziali della modellazione di una serie storica di osservazioni tramite hidden Markov models sono molteplici:

1. stima in tempo reale dello stato latente al tempo t date le osservazioni $\boldsymbol{y}^{(t)}$ ("filtering");

2. previsione di un'osservazione futura y_{t+h} dal processo, conoscendo $\boldsymbol{y}^{(t)}$;

3. sempre avendo a disposizione l'intera sequenza di osservazioni $\boldsymbol{y}^{(t)}$, stima retrospettiva:

   — di un singolo stato latente ("smoothing" o "local decoding");

   — della sequenza più verosimile di stati latenti ("global decoding").

Stima dei parametri

Siano $y_1, \ldots, y_t$ le realizzazioni di un hidden Markov model con m stati latenti e distribuzione iniziale $\boldsymbol{\upsilon}$. L'obiettivo è quello di stimare i parametri del modello date le $\boldsymbol{y}^{(t)}$, massimizzando la verosimiglianza $L_t(\boldsymbol{\theta})$:

$$L_t(\boldsymbol{\theta}) = L_t(\boldsymbol{\theta}; \boldsymbol{y}^{(t)}) = \mathbb{P}(\boldsymbol{Y}^{(t)} = \boldsymbol{y}^{(t)}) = \boldsymbol{\upsilon}\boldsymbol{P}(y_1)\boldsymbol{\Gamma}\boldsymbol{P}(y_2)\boldsymbol{\Gamma}\boldsymbol{P}(y_3)\cdots\boldsymbol{\Gamma}\boldsymbol{P}(y_t)\mathbf{1}', \quad (3.9)$$

dove $\boldsymbol{\Gamma}$ è la t.p.m. definita nella (3.3), $\mathbf{1}' = (1, 1, \ldots, 1)^T$ e le $\boldsymbol{P}(y_i), i = 1, \ldots, t$, sono matrici diagonali così strutturate:

$$\begin{pmatrix} p_1(y_i) & & 0 \\ & \ddots & \\ 0 & & p_m(y_i) \end{pmatrix}. \quad (3.10)$$

I parametri del modello sono:

— le probabilità di transizione da uno stato all'altro della catena di Markov;

— quelli della distribuzione iniziale degli stati latenti $\boldsymbol{\upsilon} = (\upsilon_1, \ldots, \upsilon_m)$;

— quelli della distribuzione state-dependent.

Il numero m di stati latenti è invece una costante fissata. Al crescere del numero di stati latenti m, cresce il valore della verosimiglianza. Tuttavia, al netto di eventuali assunzioni semplificatrici sulla struttura di $\boldsymbol{\Gamma}$, all'aumentare unitario di m cresce in maniera quadratica il numero di parametri da stimare: è opportuno basare la scelta tra due diversi HMMs su criteri di informazione automatica come AIC (Criterio di informazione di Akaike) e BIC (Criterio di informazione Bayesiano o di Schwartz), che penalizzano i modelli più complicati.

La funzione di verosimiglianza (3.9) è non lineare in $\boldsymbol{\theta}$ ed è multimodale, dunque presenta vari massimi locali. Per trovare le stime di massima verosimiglianza, è conveniente l'uso dell'algoritmo Expectation-Maximization (EM), che nel contesto degli HMMs viene anche detto di Baum-Welch dal nome dei suoi inventori. Intuitivamente, con l'algoritmo di Baum-Welch si considerano gli stati latenti della catena di Markov come dati mancanti. A partire da un valore iniziale $\boldsymbol{\theta}_0$ dei parametri di interesse, nel passo di Expectation si calcolano i valori attesi condizionati dei dati mancanti date le osservazioni e le stime correnti di $\boldsymbol{\theta}$; nel passo di Maximization, si massimizza rispetto a $\boldsymbol{\theta}$ la funzione di verosimiglianza con i dati mancanti rimpiazzati dai valori attesi calcolati al passo precedente. Avvalendosi di aggiornamenti iterativi per migliorare la verosimiglianza in modo incrementale, l'algoritmo EM è sensibile alla scelta dei valori utilizzati per la sua inizializzazione, e può convergere a massimi locali anziché al massimo globale. Nella Sezione 3.2, un grafico mostrerà

empiricamente i risultati delle stime ottenute inizializzando l'algoritmo da punti di partenza diversi.

Filtering

Indichiamo l'insieme delle osservazioni $(y_1, \ldots, y_t)$ in maniera sintetica con il simbolo $\boldsymbol{y}^{(t)}$. Per il problema di filtering è stato sviluppato il cosiddetto "forward algorithm", a partire dai concetti di variabile forward,

$$\alpha_t(j) = \mathbb{P}(X_t = j; y_1, \ldots, y_t) = \mathbb{P}(X_t = j; \boldsymbol{y}^{(t)}), \tag{3.11}$$

e di filtro:

$$\phi_{t|t}(j) = \mathbb{P}(X_t = j|\boldsymbol{y}^{(t)}). \tag{3.12}$$

La variabile forward esprime la probabilità che la catena di Markov sia nello stato j al tempo t, data la sequenza di osservazioni da 1 a t. È immediato notare come

$$\phi_{t|t}(j) = \mathbb{P}(X_t = j|\boldsymbol{y}^{(t)}) = \frac{\mathbb{P}(X_t = j; \boldsymbol{y}^{(t)})}{\mathbb{P}(\boldsymbol{y}^{(t)})} = \frac{\alpha_t(j)}{L_t}, \tag{3.13}$$

risultato che deriva dall'applicazione della legge delle probabilità totali a denominatore. Le variabili forward $\alpha_t(j)$ possono essere ricavare ricorsivamente a partire da

$$\alpha_1(j) = \mathbb{P}(X_1 = j; y_1) = \mathbb{P}(y_1|X_1 = j)\mathbb{P}(X_1 = j) = g(y_1|j)\upsilon_j, \tag{3.14}$$

dove con υ_j indichiamo la probabilità iniziale associata allo stato latente j. La formula alla base del forward algorithm è la seguente:

$$\begin{aligned}\alpha_t(j) = \mathbb{P}(X_t = j; \boldsymbol{y}^{(t)}) &= \sum_{i=1}^{m} \mathbb{P}(X_t = j, X_{t-1} = i; \boldsymbol{y}^{(t)}) \\ &= \sum_{i=1}^{m} \mathbb{P}(y_t|X_t = j, X_{t-1} = i; \boldsymbol{y}^{(t-1)}) \\ &\quad \times \mathbb{P}(X_t = j|X_{t-1} = i; \boldsymbol{y}^{(t-1)}) \\ &\quad \times \mathbb{P}(X_{t-1} = i; \boldsymbol{y}^{(t-1)})\end{aligned} \tag{3.15}$$

Possiamo semplificare la (3.15) notando che:

— $\mathbb{P}(y_t|X_t = j, X_{t-1} = i; \boldsymbol{y}^{(t-1)}) \equiv \mathbb{P}(y_t|X_t = j) = g(y_t|j)$,
perché X_{t-1} ha effetto solo su X_t e le $\boldsymbol{y}^{(t-1)}$ a loro volta non influenzano y_t, in quanto legate alle $\boldsymbol{x}^{(t-1)} = (x_1, \ldots, x_{t-1})$;

— $\mathbb{P}(X_t = j|X_{t-1} = i; \boldsymbol{y}^{(t-1)}) \equiv \mathbb{P}(X_t = j|X_{t-1} = i) = \gamma_{ij}(1)$
perché $\boldsymbol{y}^{(t-1)}$ contiene informazioni irrilevanti, è sufficiente la conoscenza della realizzazione di X_{t-1};

— $\mathbb{P}(X_{t-1} = i; \boldsymbol{y}^{(t-1)}) = \alpha_{t-1}(i)$
per definizione di variabile forward.

Utilizzando le semplificazioni descritte, perveniamo ad una riscrittura della (3.15):

$$\alpha_t(j) = \mathbb{P}(X_t = j; \boldsymbol{y}^{(t)}) = \sum_{i=1}^{m} \alpha_{t-1}(i)\gamma_{ij}(1)g(y_t|j). \tag{3.16}$$

Purtroppo, la sequenza $(\alpha_t)_{t\geq 1}$ va a 0 o a $+\infty$ quando $t \to +\infty$ con velocità esponenziale. Una soluzione è scalare, ad ogni step dell'algoritmo, la variabile forward $\alpha_t(j)$, dividendola per un coefficiente indipendente dal particolare j, ma dipendente da t: $c_t = \sum_{i=1}^{m} \alpha_t(i)$. Otteniamo esattamente il filtro (3.12), che rappresenta dunque una variabile forward stabilizzata.

Previsione

La previsione ad orizzonte temporale k dello stato latente della catena di Markov di un hidden Markov model è

$$\phi_{t+k|t}(j) = \mathbb{P}(X_{t+k} = j|\boldsymbol{y}^{(t)}) \tag{3.17}$$

e coincide, applicando ancora una volta la legge delle probabilità totali, con

$$\phi_{t+k|t}(j) = \sum_{i=1}^{m} \mathbb{P}(X_{t+k} = j|X_t = i; \boldsymbol{y}^{(t)}) \underbrace{\mathbb{P}(X_t = i|\boldsymbol{y}^{(t)})}_{\phi_{t|t}(i)}. \tag{3.18}$$

Nella (3.18), notiamo che il condizionamento a $\boldsymbol{y}^{(t)}$ nel primo fattore è irrilevante nel determinare la probabilità che X assuma lo stato j al tempo $t+k$. La succitata equazione diventa, allora,

$$\begin{aligned} \phi_{t+k|t}(j) &= \sum_{i=1}^{m} \underbrace{\mathbb{P}(X_{t+k} = j|X_t = i)}_{\substack{\text{probabilità di transizione} \\ \text{dopo } k \text{ passi}}} \phi_{t|t}(i) \\ &= \sum_{i=1}^{m} \gamma_{ij}(k)\phi_{t|t}(i) \end{aligned} \tag{3.19}$$

Evidenziamo che, per $k \to +\infty$, $\gamma_{ij}(k) \to \delta_j$: le probabilità di transizione da un qualsiasi stato latente al j-esimo convergono alla probabilità di osservare j nella

distribuzione stazionaria. Da ciò deriva che, sempre per valori grandi di k,

$$\phi_{t+k|t}(j) = \sum_{i=1}^{m} \delta_j \phi_{t|t}(i) = \delta_j \underbrace{\sum_{i=1}^{m} \phi_{t|t}(i)}_{=1} = \delta_j. \tag{3.20}$$

A questo punto, a partire dai risultati appena ottenuti sulla previsione dello stato latente, possiamo ricavare la previsione della distribuzione condizionata alle osservazioni $\boldsymbol{y}^{(t)}$ di Y_{t+k}.

$$\begin{aligned} \mathbb{P}(Y_{t+k}|\boldsymbol{y}^{(t)}) &= \sum_{i=1}^{m} \mathbb{P}(Y_{t+k}|X_{t+k} = i; \boldsymbol{y}^{(t)}) \mathbb{P}(X_{t+k} = i|\boldsymbol{y}^{(t)}) \\ &= \sum_{i=1}^{m} g(y_{t+k}|i) \phi_{t+k|t}(i) \end{aligned} \tag{3.21}$$

Smoothing (o local decoding)

Nello smoothing, lo scopo diventa quello di trovare la probabilità di un certo stato latente al tempo t, avendo collezionato anche osservazioni dopo t:

$$\phi_{t|T}(i) = \mathbb{P}(X_t = i|\boldsymbol{y}^{(T)}), \quad 1 \leq t \leq T. \tag{3.22}$$

Per le proprietà della probabilità condizionata, $\phi_{t|T}(i)$, che viene chiamato smoother, è proporzionale alla congiunta

$$\begin{aligned} \mathbb{P}(X_t = i; \boldsymbol{y}^{(T)}) &= \mathbb{P}(X_t = i; \boldsymbol{y}^{(t)}, y_{t+1:T}) \\ &= \mathbb{P}(y_{t+1:T}|X_t = i; \boldsymbol{y}^{(t)}) \mathbb{P}(X_t = i; \boldsymbol{y}^{(t)}) = \beta_t(i)\alpha_t(i) \end{aligned} \tag{3.23}$$

Le quantità $\beta_t(i) = \mathbb{P}(y_{t+1:T}|X_t = i; \boldsymbol{y}^{(t)})$ sono dette variabili backward, ed esprimono la probabilità di osservare la sequenza $y_{t+1:T}$ avendo osservato $\boldsymbol{y}^{(t)}$ e lo stato i al tempo t. Lo smoother, in funzione di variabili backward e forward, diventa

$$\phi_{t|T}(i) = \frac{\alpha_t(i)\beta_t(i)}{L_T} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{m} \alpha_t(j)\beta_t(j)}. \tag{3.24}$$

dove L_t è la funzione di verosimiglianza. Inoltre, si può dimostrare che

$$\beta_t(i) = \sum_{j=1}^{m} \beta_{t+1}(j) g(y_{t+1}|j) \gamma_{ij}(t). \tag{3.25}$$

La procedura induttiva con la quale, a partire da $\beta_T(i) = 1$, si calcolano per $t =$

$T-1,\dots,1$ i β_t è affetta dagli stessi problemi del forward algorithm. Per questo motivo, ad ogni step i $\beta_t(i)$ vengono normalizzati dividendoli per $d_t = \sum_{j=1}^{m} \beta_t(j)$.

Global decoding

In alcune applicazioni non è tanto di interesse stimare lo stato latente al tempo t, quanto piuttosto stimare l'intera sequenza degli stati latenti dato l'insieme delle osservazioni cumulate $\boldsymbol{y}^{(t)}$. Anziché applicare lo smoother (3.24) per ogni t, si può massimizzare la probabilità condizionata $\mathbb{P}(\boldsymbol{X}^{(t)} = \boldsymbol{x}^{(t)} | \boldsymbol{Y}^{(t)} = \boldsymbol{y}^{(t)})$. Le tecniche disponibili sono due:

1. algoritmo brute-force che calcola la suddetta probabilità per ogni sequenza possibile di stati. Chiaramente, essendo le possibili permutazioni delle sequenze pari a m^t, questo metodo non è applicabile per elevati valori di m e/o di t;

2. algoritmo di Viterbi. Nato negli Anni Sessanta, questo algoritmo utilizza tecniche di programmazione dinamica per calcolare il cammino ("path") delle sequenze più plausibile in modo efficiente.

3.1.4 Possibili estensioni del modello

La forma base del modello hidden Markov, specificata nella (3.6), si presta a varie estensioni. Quelle più comunemente utilizzate sono di quattro tipologie, che ci limitiamo a descrivere nei loro tratti più rilevanti.

Inclusione di covariate

È comune introdurre un vettore di covariate z_t nel modello, ad esempio componenti di trend e stagionalità modellate come funzioni parametriche del tempo. In particolare, l'effetto delle covariate può essere incorporato nella matrice delle probabilità di transizione $\boldsymbol{\Gamma}$. Ciò porta alla perdita della proprietà di omogeneità (3.2); in ogni caso, le righe della matrice $\boldsymbol{\Gamma}$ devono rispettare il vincolo di somma a 1. L'alternativa è di far dipendere dalle variabili esplicative la media della variabile state-dependent Y_t, $E[Y_t | X_t = i]$, stimando parametri diversi per ogni stato latente. Per HMMs con tre o più stati latenti, generalmente è più semplice e computazionalmente meno onerosa quest'ultima soluzione.

In generale, al netto dei problemi computazionali, dovremmo basare la scelta sull'ipotesi di lavoro fatta a monte. Se questa prevede che le covariate influiscano sui tassi di passaggio da uno stato all'altro, è meglio esprimere per mezzo delle z_t i parametri della matrice della probabilità di transizione; al contrario, se è più

sensato supporre che le covariate non influiscano su tali tassi, allora sono uno o più parametri della distribuzione state-dependent a dover essere messi in funzione di z_t.

HMM per dati longitudinali

Un'ulteriore estensione degli HMMs è stata sviluppata per lavorare in presenza di dati panel o longitudinali, nella situazione in cui K serie storiche dello stesso tipo, dette "serie componenti", sono rilevate su K unità statistiche diverse (soggetti, stazioni meteorologiche, ecc.). I dataset costruiti per la presente tesi sono proprio serie storiche multivariate collezionate per varie città venete, europee ed africane.

In certe applicazioni, è conveniente o sensato ipotizzare che a guidare tutte le K serie componenti sia la stessa sequenza di stati latenti. Guttorp & Zucchini (1991), ad esempio, modellano in questo modo l'occorrenza di giorni con e senza pioggia in alcuni siti delle Grandi Pianure del Nord America; tuttavia, questa sincronia tra stati latenti molto spesso è irrealistica.

Un altro approccio è quello di imporre totale indipendenza tra le K serie storiche univariate. Così facendo, le sequenze di stati latenti e i parametri della distribuzione state-dependent, al termine della procedura di stima, risultano diversi da serie a serie. Se, chiamando ancora in causa il lavoro di Guttorp e Zucchini, modellassimo l'eventualità della pioggia in località molto distanti tra di loro, anziché tutte concentrate nelle Grandi Pianure, quest'ultimo modo di procedere sarebbe più ragionevole. Tuttavia, in presenza di molte covariate di cui tener conto o di molti stati latenti, il metodo "no pooling" pecca di alto costo computazionale.

C'è poi la possibilità di mettere in atto un "pooling parziale" delle informazioni provenienti dalle serie componenti, ammettendo l'uguaglianza tra alcuni (o tutti) i parametri delle distribuzioni state-dependent e/o tra quelli delle matrici delle probabilità di transizione. Di solito, il pooling parziale porta a modelli più parsimoniosi in virtù del ridotto numero di parametri rispetto al caso di indipendenza tra le serie componenti, nonché ad errori standard più bassi.

Catene di Markov di ordine superiore al primo

L'ordine della catena di Markov alla base di un HMM può essere aumentato. La proprietà Markoviana di una generica catena di ordine l diventa

$$\mathbb{P}(X_t|\boldsymbol{X}^{(t-1)}) = \mathbb{P}(X_t|X_{t-1},\dots,X_{t-l}). \quad \forall t = 1,2,\dots \tag{3.26}$$

Rispetto alla (3.1), viene rilassata l'assunzione che il modello abbia memoria limitata al ritardo 1. Per $l = 2$, le probabilità di transizione $\gamma_{ij}(t)$ della (3.2) diventano

$$\gamma_{ijk}(t) = \mathbb{P}(X_t = k|X_{t-1} = j, X_{t-2} = i) \tag{3.27}$$

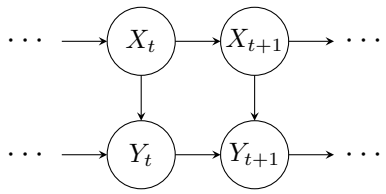


Figura 3.2: Rappresentazione grafica della struttura di dipendenza presente in un modello Markov-switching AR(1).

e perciò, soprattutto per un numero m di stati latenti molto grande, il numero di parametri aumenta velocemente.

Modelli Markov-switching

Una rilevante estensione dei modelli hidden Markov prevede l'aggiunta di una dipendenza tra le variabili del processo che genera le osservazioni. Questa dipendenza può coinvolgere osservazioni successive (Figura 3.2; confrontare con Figura 3.1) o anche osservazioni a ritardi temporali più ampi. Il modello risultante è detto Markov-switching di Hamilton (1989) ed è molto usato in ambito economico e finanziario per analizzare serie temporali che mostrano cambiamenti strutturali o regimi differenti nel tempo.

3.1.5 Hidden semi-Markov models

Come spiegato nel Paragrafo 3.1.4, una possibile estensione dell'HMM prevede l'utilizzo di catene Markoviane con ordine $l > 1$. Esiste un modo alternativo e più parsimonioso per aggirare l'assunzione di memoria pari a un ritardo negli HMMs. Vale la pena notare, come prima cosa, che il tempo trascorso in un certo stato dalla catena di Markov del prim'ordine di un HMM — chiamato "dwell time" o "sojourn time" e indicato con d_i — segue una distribuzione geometrica:

$$d_i(r) = (1 - \gamma_{ii})\gamma_{ii}^{r-1} \tag{3.28}$$

La (3.28) implica che la probabilità di rimanere nello stato i diminuisce esponenzialmente man mano che aumenta il tempo trascorso nello stesso stato. La classe degli hidden semi-Markov models (HSMMs) consente di evitare questa situazione specificando una qualsiasi altra distribuzione sul tempo di permanenza in uno stato. Per questo motivo, gli HSMMs sono anche chiamati "explicit duration HMMs" o "variable-duration HMMs" (Yu, 2010).

Processi semi-Markoviani

Sia $(C_t)_{t\geq 1}$ una catena di Markov omogenea del prim'ordine, avente state space $\{1, 2 \dots, m\}$ e matrice delle probabilità di transizione $\mathbf{\Omega}$ tale che

$$\Omega_{11} = \Omega_{22} = \cdots = \Omega_{mm} = 0. \tag{3.29}$$

La caratteristica (3.29) fa sì che, in una realizzazione dalla catena di Markov con una siffatta matrice delle probabilità di transizione, non vi possano essere due valori consecutivi uguali. Supponiamo poi che da ogni elemento i di C_t — che in questo contesto prende il nome di "embedded Markov Chain" — si generi una sequenza di valori uguali a i stesso, con lunghezza realizzazione di d_i (il tempo di permanenza se $C_t = i$); e che questi nuovi valori, insieme, formino una sequenza X_t^S. Ad esempio, se $m = 3$ e $C_t = \{2, 3, 2, 1, 3, \dots\}$, una possibile realizzazione di X_t^S è $\{22\ 333\ 222\ 1111\ 33 \dots\}$, dove la sequenza 1111 è realizzazione di d_1 e le due sequenze 22 e 222 sono realizzazioni indipendenti di d_2, così come 333 e 33 lo sono di d_3.

Un processo X_t^S con tali caratteristiche non è — a meno che le distribuzioni dei d_i siano tutte geometriche — un processo di Markov, ma un processo semi-Markoviano, con probabilità di permanenza in un certo stato dettate dai d_i. La distribuzione di questi ultimi può essere una qualsiasi che abbia come supporto $\mathbb{Z}^+$ (ad esempio, una Poisson troncata in zero).

HMM come approssimazione dell'HSMM

Un hidden semi-Markov model si compone di un processo osservabile $(y_t^S)_{t\geq 1}$ e di un processo semi-Markoviano latente del tipo descritto al paragrafo precedente. Questo modello può essere rappresentato approssimativamente come un particolare hidden Markov model avente uno state space espanso; ciò consente, tra l'altro, l'estensione delle teorie inferenziali viste per gli HMMs agli HSMMs. L'idea è quella di ottenere una rappresentazione del tempo di permanenza nello stato i dell'HSMM come tempo di permanenza nello "stato aggregato" I_i di un HMM approssimante.

Sia nuovamente X_t^S un processo semi-Markoviano che genera osservazioni y_t^S. La sua embedded Markov Chain C_t ha una matrice delle probabilità di transizione $\mathbf{\Omega}$ tale per cui

$$\omega_{ij} = \mathbb{P}(C_{t+1} = j | C_t = i, C_{t+1} \neq i), \quad i, j = 1, \dots, m, \tag{3.30}$$

con $w_{ii} = 0\ \forall i = 1, \dots, m$ e $\sum_{j=1}^m w_{ij} = 1$. Siano, poi, $m_1, m_2, \dots, m_m$ interi positivi; poniamo $m_0 = 0$. Consideriamo inoltre un HMM con Y_t governate da una catena di Markov X_t definita nello state space espanso $\{1, 2, \dots, \sum_{i=1}^m m_i\}$, atto ad approssimare l'hidden semi-Markov model. Definiamo le quantità I_k come

aggregazioni degli stati di X_t, una per ogni elemento dello state space di X_t^S:

$$I_k = \left\{ n : \sum_{i=0}^{k-1} m_i < n \leq \sum_{i=0}^{k} m_i \right\}, \quad k = 1, \ldots, m. \tag{3.31}$$

Le I_k racchiudono tutti gli stati compresi tra un m_i e il successivo. Un'assunzione chiave è che la distribuzione di Y_t dato $X_t = l$, $l \in I_i$, sia la stessa per tutti gli altri stati appartenenti ad una medesima aggregazione I_i, e anche la stessa di Y_t^S condizionatamente a $X_t^S = i$:

$$p(Y_t | X_t \in I_i) = p(Y_t^S | X_t^S = i), \quad t = 1, \ldots, T;\ i = 1, \ldots, m. \tag{3.32}$$

La matrice delle probabilità di transizione di X_t, la catena di Markov dell'HMM, si può partizionare in $m \times m$ blocchi come nella (3.33):

$$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Gamma}_{11} & \ldots & \boldsymbol{\Gamma}_{11} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Gamma}_{m1} & \ldots & \boldsymbol{\Gamma}_{mm} \end{pmatrix} \tag{3.33}$$

dove ogni entrata diagonale è un blocco di dimensioni $m_i \times m_i$ definito, per $m_i \geq 2$, come

$$\boldsymbol{\Gamma}_{ii} = \begin{pmatrix} 0 & 1 - c_i(1) & 0 & \ldots & 0 \\ \vdots & 0 & \ddots & & \vdots \\ & \vdots & & & 0 \\ 0 & 0 & \ldots & 0 & 1 - c_i(m_i - 1) \\ 0 & 0 & \ldots & 0 & 1 - c_i(m_i) \end{pmatrix}, \tag{3.34}$$

e ogni blocco fuori dalla diagonale (di dimensioni $m_i \times m_j$) come

$$\boldsymbol{\Gamma}_{ij} = \begin{pmatrix} w_{ij} c_i(1) & 0 & \ldots & 0 \\ w_{ij} c_i(2) & 0 & \ldots & 0 \\ \vdots & & & \\ w_{ij} c_i(m_i) & 0 & \ldots & 0 \end{pmatrix}. \tag{3.35}$$

Per $m_i = 1$, $\boldsymbol{\Gamma}_{ii} = 1 - c_i(1)$, e per $m_j = 1$ le colonne di zeri spariscono dalla 3.35. I $c_i(r)$, detti "hazard rate" delle distribuzioni di d_i (Langrock & Zucchini, 2011), sono definiti, per $r = 1, 2, 3, \ldots, m_i$, dalla (3.36):

$$c_i(r) = \frac{d_i(r)}{1 - F_i(r-1)} \tag{3.36}$$

e dipendono così solo dai parametri delle distribuzioni dei tempi di permanenza; il che vuol dire che differenti dwell time portano a differenti c_i, ma la struttura di $\mathbf{\Gamma}$ rimane inalterata.

Interpretazione dell'approssimazione

La matrice delle probabilità di transizione (3.33) è quella alla base dell'HMM che approssima l'HSMM. Tutte le transizioni entro un certo una certa aggregazione di stati I_i sono governate dal blocco $\mathbf{\Gamma}_{ii}$ sulla diagonale, che dunque determina la distribuzione del tempo di permanenza nell'aggregazione stessa — ovvero, la distribuzione del tempo di permanenza in uno stato nell'HSMM. Invece, i blocchi fuori dalla diagonale di $\mathbf{\Gamma}$ regolano le probabilità di transitare da un'aggregazione all'altra, e cioè da uno stato all'altro dell'HSMM.

È possibile dimostrare che, per $i \neq j, 1 \leq i, j \leq m$,

$$\mathbb{P}(X_{t+1} \in I_j | X_t \in I_i, X_{t+1} \notin I_i) = w_{ij}. \tag{3.37}$$

Questo risultato, insieme alla (3.32), implica che l'HMM creato è capace di rappresentare, almeno approssimativamente, una qualsiasi distribuzione del tempo di permanenza di un HSMM a m stati; l'approssimazione è tanto più precisa quanto più grandi sono i m_i.

HSMMs e previsioni meteorologiche

Un hidden Markov model è strettamente vincolato alla proprietà Markoviana (3.1), secondo cui lo stato corrente del sistema dipende solo da quello precedente. Pertanto, un HMM ha normalmente una memoria a breve termine della storia passata: ciò può costituire un limite quando si tratta di prevedere eventi futuri. Negli HSMM, si supera questo problema: l'indipendenza condizionale tra passato e futuro è garantita solo quando il processo passa da uno stato ad uno stato distinto (anziché ad ogni passo temporale come nel caso Markoviano classico).

Nell'ambito della meteorologia, gli HSMMs possono gestire sequenze temporali con variazioni di lunghezza più dinamiche rispetto agli HMMs. Questo è importante, poiché le sequenze di eventi atmosferici possono variare notevolmente in termini di durata e intensità. In letteratura, troviamo alcuni esempi di applicazione degli hidden semi-Markov models a dati meteorologici nei lavori di Dong et al. (2009), che hanno proposto un hidden semi-Markov model per la previsione della concentrazione di PM2.5 nell'Illinois, e di Sansom & Thomson (2001), che hanno modellato i tassi di precipitazione di Wellington (Nuova Zelanda).

3.2 Applicazioni

I modelli hidden Markov sono strumenti di analisi di serie storiche molto versatili. I due dataset ricavati dal database di ERA5 si prestano, potenzialmente, a numerosissime modellazioni; in questa Sezione ne proponiamo alcune. Il pacchetto R che abbiamo utilizzato allo scopo è `depmixS4` (Visser & Speekenbrink, 2010), consigliato da Zucchini et al. (2016).

3.2.1 Serie storica univariata, dati giornalieri

Come primo esempio di applicazione del modello hidden Markov ai dati meteorologici, consideriamo come variabile risposta la temperatura massima giornaliera registrata dal 2010 al 2023 (temperature_2m_max) nella città di Vienna. Qualsiasi serie storica giornaliera di temperature, siano esse massime o minime, mostra una forte componente di stagionalità, nel senso che tende a ripetersi in maniera pressoché analoga di anno in anno a causa, naturalmente, dell'avvicendarsi delle stagioni climatiche. Stando a quanto osservato in fase di analisi esplorativa, e precisamente in Figura 2.5, oltre alla stagionalità è presente anche un trend crescente di fondo di cui tener conto.

Trasformazione delle variabili

Prima di procedere con l'adattamento del modello per le temperature massime di Vienna, effettuiamo alcune operazioni preliminari di modifica delle variabili. Per incorporare la stagionalità, dopo aver estratto dalla variabile date l'informazione su anno, mese e giorno, creiamo una variabile season con livelli inverno (22 dicembre – 20 marzo), primavera (21 marzo – 21 giugno), estate (22 giugno – 22 settembre) e autunno (23 settembre – 21 dicembre). In più, per favorire una migliore interpretazione dei coefficienti, alla variabile year sottraiamo l'anno di inizio della serie storica (2010). Il dummy coding applicato ai mesi dell'anno, raggruppati in stagioni, ha come pregio la semplicità interpretativa, però ignora la natura ciclica dei dati (nel senso che le stagioni non vengono viste come una consecutiva all'altra). Un altro approccio comune per codificare variabili categoriche cicliche, come i mesi dell'anno o le ore del giorno, si avvale di trasformazioni trigonometriche e verrà discusso al Paragrafo 3.2.2.

La variabile wind_direction_10m_dominant, che esprime la direzione del vento in gradi sessagesimali, viene ricodificata in un fattore a otto livelli secondo la rosa dei venti: N/NE (0°–45°), E/NE (45°–90°), E/SE (90°–135°), S/SE (135°–180°), S/SW (180°–225°), W/SW (225°–270°), W/NW (270°–315°), N/NW (315°–360°). La ricodifica è sicuramente preferibile all'inserire in un modello lineare la variabile

Tabella 3.1: Log-verosimiglianza, AIC e BIC dei modelli con m stati latenti applicati alla serie delle temperature massime giornaliere di Vienna.

	Log-verosimiglianza	AIC	BIC
$m = 2$	-11840.5	23775.0	24082.3
$m = 3$	-11389.8	22927.6	23411.4
$m = 4$	-11164.0	22534.1	23207.5
$m = 5$	-11045.5	22358.9	23235.0
$m = 6$	-10936.5	22207.0	23298.8

wind_direction_10m_dominant senza trasformazioni, ma anche in questo caso distrugge la vicinanza tra direzioni del vento provenienti dallo stesso quadrante e fa aumentare il numero di parametri. Oltre a ciò, a partire dalle variabili sunrise e sunset, ricaviamo sun_hours che esprime il tempo in ore tra alba e tramonto. Infine, dividiamo la variabile sunshine_duration per 3600 onde renderla nella stessa scala di sun_hours.

Scelta del numero di stati latenti

La media della distribuzione state-dependent, che possiamo considerare come Normale, è dunque modellata nel modo seguente:

$$\begin{aligned}\mu = \beta_0 &+ \beta_1 \cdot \mathsf{I(year{-}2010)} + \boldsymbol{\beta}_2 \cdot \mathsf{season} + \beta_3 \cdot \mathsf{rain\_sum} \\ &+ \beta_4 \cdot \mathsf{precipitation\_hours} + \beta_5 \cdot \mathsf{wind\_speed\_10m\_max} \\ &+ \beta_6 \cdot \mathsf{wind\_gusts\_10m\_max} + \boldsymbol{\beta}_7 \cdot \mathsf{wind\_direction\_10m\_dominant} \\ &+ \beta_8 \cdot \mathsf{shortwave\_radiation\_sum} + \beta_9 \cdot \mathsf{et0\_fao\_evapotranspiration} \\ &+ \beta_{10} \cdot \mathsf{snowfall\_sum} + \beta_{11} \cdot \mathsf{sun\_hours} + \beta_{12} \cdot \mathsf{sunshine\_duration},\end{aligned} \tag{3.38}$$

dove $\boldsymbol{\beta}_2$ e $\boldsymbol{\beta}_7$ sono in grassetto ad indicare che per le variabili categoriali season e wind_direction_10m_dominant sono stimati più coefficienti (uno in meno del numero totale di livelli della variabile).

Sono appositamente scartati dall'insieme di stima i dati relativi agli ultimi 10 giorni del 2023: verranno utilizzati come insieme di verifica per mettere alla prova il modello. Ovviamente, abbandoniamo la convenzione, usata nel machine learning, di dividere il dataset assegnando casualmente il 75% delle osservazioni all'insieme di stima e il rimanente 25% all'insieme di verifica. Questa è la prassi quando si lavora con serie storiche; d'altro canto, in ambito meteorologico è decisamente più interessante realizzare previsioni per osservazioni future rispetto all'ultima misurazione che non per quelle di giorni passati scelti in modo arbitrario.

Tabella 3.2: Matrice delle probabilità di transizione dell'HMM applicato alla serie univariata delle temperature massime giornaliere di Vienna.

	St.1	St.2	St.3	St.4
St.1	0.800	0.008	0.192	$\simeq 0$
St.2	$\simeq 0$	0.694	0.239	0.066
St.3	0.144	0.125	0.719	0.012
St.4	0.009	0.089	$\simeq 0$	0.902

Come prima cosa, effettuiamo una scelta del numero m di stati latenti basata sul BIC. La Tabella 3.1 mostra i valori di log-verosimiglianza, AIC e BIC per modelli con un numero di stati latenti che va da 2 a 6. Basando la scelta del modello sul BIC, risulta preferibile quello con $m = 4$ stati latenti; i valori dell'AIC, invece, esibiscono una decrescita monotona. Ricordiamo che l'AIC non è consistente, ovvero ha probabilità non nulla di portare alla selezione di un modello sovra-parametrizzato.

Interpretazione dei risultati

Dalla stima, effettuata grazie al pacchetto `depmixS4` di R, otteniamo la matrice delle probabilità di transizione tra stati di Tabella 3.2 e i parametri elencati in Tabella 3.3. Dall'analisi della t.p.m., notiamo che la persistenza nei vari stati è particolarmente alta per lo stato latente 4, mentre dallo stato 2 è relativamente facile traslare allo stato 3. Per converso, la stima di certe probabilità di transizione — come quella dallo stato 1 allo stato 4, o dallo stato 2 allo stato 1 — è praticamente nulla.

L'interpretazione dei parametri è la stessa che solitamente facciamo per qualsiasi modello lineare, solo che nel caso del modello hidden Markov c'è una stima per ognuno degli stati latenti. Uno sguardo alla Tabella 3.3 rivela, ad esempio, come venti dalle direzioni N/NW (nord/nord-ovest) e N/NE (nord/nord-est, la categoria di riferimento) portino ad un calo della temperatura in tutti gli stati latenti. Il coefficiente $\hat{\beta}_{10}$, quello relativo alla variabile snowfall_sum, è negativo per tutti gli stati tranne che per il quarto. Ciò non deve sorprendere eccessivamente, perché spesso, in inverno, la copertura nuvolosa che origina le precipitazioni è in grado di limitare la dispersione termica mantenendo il calore nei bassi strati dell'atmosfera.

Per dare un'interpretazione degli stati latenti, possiamo provare ad utilizzare la variabile weather_code, che non è stata inclusa nel modello, come variabile di stratificazione. I livelli di quest'ultima vengono accorpati come segue (tra parentesi i codici WMO originali): Poco nuvoloso (0, 1, 2), Coperto (3), Pioggia debole (51, 53, 55), Pioggia (71, 73, 75) e Neve (71, 73, 75). La Tabella di contingenza bivariata

Tabella 3.3: Coefficienti dell'HMM con 4 stati latenti applicato alla serie delle temperature massime giornaliere di Vienna.

Coef.	Variabile	St.1	St.2	St.3	St.4
$\hat{\beta}_0$	Intercetta	9.90	−0.34	4.64	−9.22
$\hat{\beta}_1$	I(year−2010)	0.06	0.02	0.03	0.19
$\hat{\boldsymbol{\beta}}_2$	season: estate	−3.62	6.84	2.47	2.43
	season: inverno	−3.97	−3.53	−3.48	−2.28
	season: primavera	−3.01	−2.99	−1.82	−4.10
$\hat{\beta}_3$	rain_sum	0.09	0.03	0.18	0.65
$\hat{\beta}_4$	precipitation_hours	0.05	0.05	−0.01	−0.06
$\hat{\beta}_5$	wind_speed_10m_max	−0.17	−0.09	−0.10	−0.20
$\hat{\beta}_6$	wind_gusts_10m_max	$\simeq 0$	−0.02	−0.02	0.04
$\hat{\boldsymbol{\beta}}_7$	wind_direction: E/NE	0.45	0.14	0.41	1.20
	wind_direction: E/SE	0.75	0.88	1.49	1.67
	wind_direction: S/NE	1.47	1.49	1.72	2.38
	wind_direction: S/SW	1.74	2.16	2.57	3.07
	wind_direction: W/SW	1.10	2.40	2.26	4.57
	wind_direction: W/NW	0.25	1.01	0.94	2.14
	wind_direction: N/NW	−0.23	−0.24	−0.01	−0.05
$\hat{\beta}_8$	shortwave_radiation_sum	−0.38	−0.42	−0.38	−0.91
$\hat{\beta}_9$	et0_fao_evapotranspiration	4.00	4.22	3.86	7.79
$\hat{\beta}_{10}$	snowfall_sum	−3.00	−0.50	−6.89	0.08
$\hat{\beta}_{11}$	sun_hours	0.56	0.82	0.61	1.12
$\hat{\beta}_{12}$	sunshine_duration	0.12	0.18	0.17	0.28

3.4 mette in luce come gli stati latenti 1 e 3, ai quali sono associati i valori più alti dell'intercetta, corrispondano soprattutto ad assenza di precipitazioni o al massimo a pioggia debole. Lo stato latente 4, invece, sembra essere di stampo invernale, essendo associato a neve e cielo coperto.

Nella Sezione 3.1.3 abbiamo visto come, per stimare la sequenza più verosimile di stati latenti, si possano usare l'approccio basato sullo smoothing oppure quello basato sull'algoritmo di Viterbi. Nella Figura 3.3, la serie storica delle temperature massime di Vienna del 2023 viene sovrapposta alla rappresentazione colorata della sequenza più verosimile di stati latenti trovata con l'algoritmo di Viterbi. I colori avorio, giallo, dorato e arancione si riferiscono, rispettivamente, agli stati 4, 2, 3 e 1. Constatiamo una predominanza degli ultimi due colori citati durante i mesi estivi; in aggiunta, il quarto stato latente si manifesta prevalentemente in inverno, coerentemente con l'interpretazione fatta in precedenza.

Tabella 3.4: Tabella di frequenza relativa bivariata tra condizione meteorologica e stati latenti dell'HMM applicato alla serie storica di Vienna.

	St.1	St.2	St.3	St.4
Poco nuvoloso	0.27	0.21	0.27	0.12
Coperto	0.20	0.29	0.22	0.32
Pioggia debole	0.34	0.32	0.35	0.20
Pioggia	0.18	0.09	0.15	0.02
Neve	0.01	0.09	0.01	0.34
	1.00	1.00	1.00	1.00

Variabilità delle stime

Nell'output fornito da `depmix()`, le stime dei coefficienti non sono accompagnate dai relativi errori standard. Per riuscire ad ottenere comunque un'idea della loro variabilità, una strategia è quella di inizializzare l'algoritmo EM (Expectation-Maximization) da punti di partenza diversi e ripetere la stima. Ciò permette, oltre che di aumentare la robustezza delle conclusioni tratte a partire dall'analisi dei coefficienti stessi, di constatare empiricamente la presenza di eventuali punti di massimo multipli qualora i valori della verosimiglianza alla convergenza siano molto diversi tra di loro. Scegliamo allora a caso 50 punti di partenza e lanciamo l'algoritmo altrettante volte.

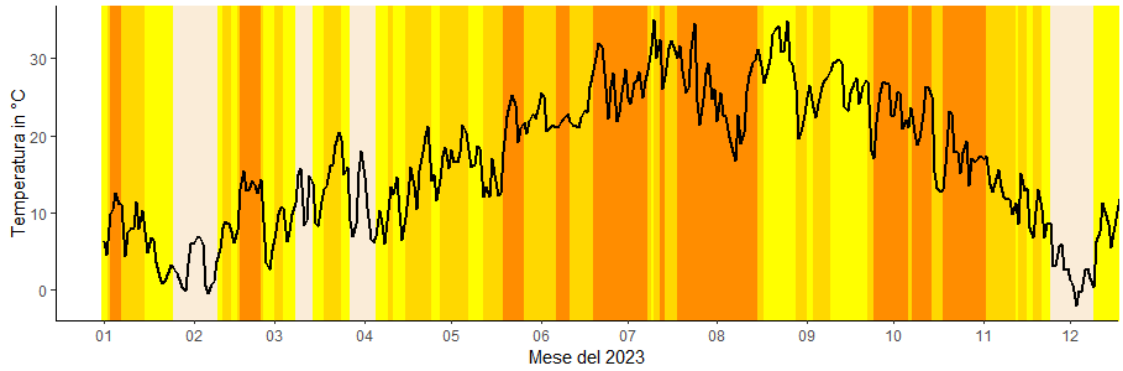


Figura 3.3: Sequenza più verosimile di stati latenti definita dall'HMM applicato alla serie delle temperature massime giornaliere di Vienna. Porzione relativa al 2023.

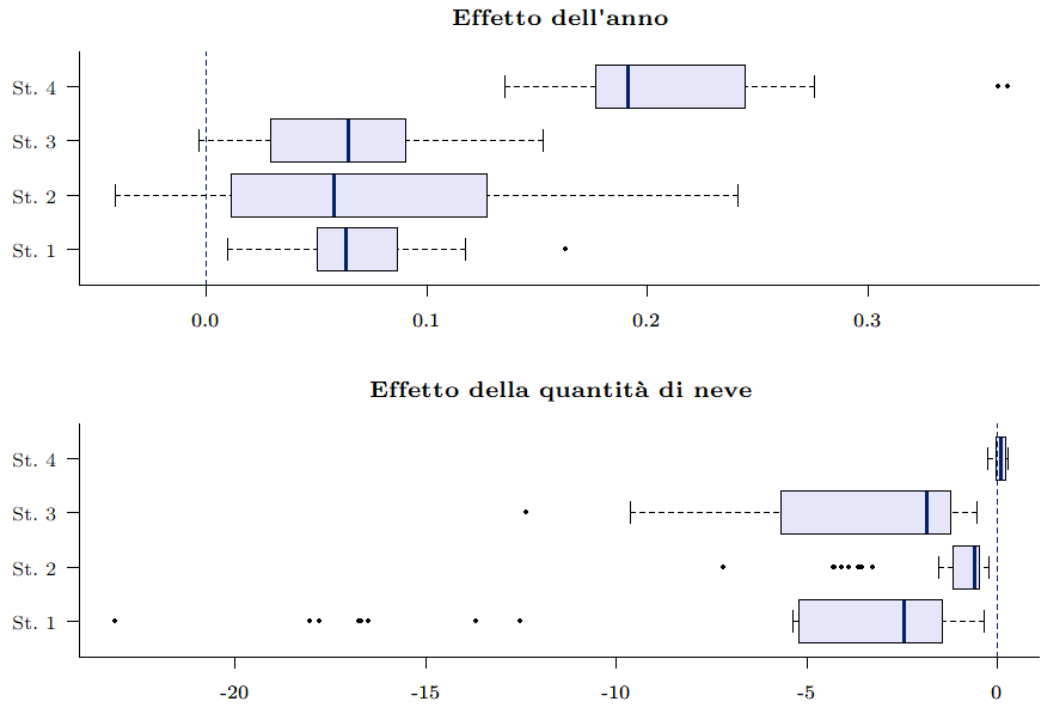


Figura 3.4: Variabilità delle stime dell'HMM applicato alla serie delle temperature giornaliere di Vienna dopo 50 ripetizioni dell'algoritmo EM.

Inizializzando l'algoritmo da punti di partenza diversi, otteniamo pressoché sempre lo stesso valore di log-verosimiglianza; per cui, ragionevolmente, possiamo affermare che quest'ultima sia concava. Nella Figura 3.4 compaiono i box plot della distribuzione delle stime di I(year−2010) e di snowfall_sum per i 4 stati latenti. La linea tratteggiata blu è posta in corrispondenza dello 0. L'effetto associato a I(year−2010) risulta essere sempre positivo per lo stato 4 e per lo stato 1, mentre si manifestano dei valori negativi per gli stati 2 e 3. Per quanto riguarda l'effetto della quantità di neve caduta, questo è praticamente nullo per lo stato latente 4 e sempre negativo per gli altri. Grafici analoghi potrebbero essere fatti per tutti gli altri coefficienti.

Il messaggio che però la Figura 3.4 comunica veramente è il seguente. Quando otteniamo delle stime tramite algoritmi iterativi dipendenti dal punto di partenza, non dobbiamo fidarci alla cieca delle indicazioni desunte dai primi valori ottenuti; la grande variabilità dei coefficienti rilevata ne è una prova. Nel contesto di una singola stima, un coefficiente particolarmente basso potrebbe essere forse "bilanciato" da valori più alti degli altri coefficienti; se ciò fa sì che la previsione finale della risposta non cambi, nella sostanza, altrettanto non possiamo dire dell'interpretazione dei

Tabella 3.5: Performance previsiva del modello hidden Markov applicato alla serie storica delle temperature massime di Vienna.

Giorno	Temperatura reale (°C)	Temperatura prevista (°C)	Scarto (°C)
22/12/2023	7.70	3.51	4.19
23/12/2023	7.10 ↘	−1.60 ↘	8.70
24/12/2023	10.90 ↗	7.60 ↗	3.30
25/12/2023	13.60 ↗	13.96 ↗	−0.36
26/12/2023	11.90 ↘	11.24 ↘	0.66
27/12/2023	10.10 ↘	6.65 ↘	3.45
28/12/2023	11.10 ↗	7.95 ↗	3.15
29/12/2023	13.50 ↗	9.74 ↗	3.76
30/12/2023	11.80 ↘	9.30 ↘	2.50
31/12/2023	4.30 ↘	5.22 ↘	−0.92

coefficienti presi individualmente.

Performance previsiva

Per testare la performance previsiva del modello, possiamo usare le 10 osservazioni relative agli ultimi giorni di dicembre 2023 contenute nell'insieme di verifica. Il pacchetto `depmixS4` di R non consente di ottenere la previsione in modo veloce: occorre costruire a mano, con un po' di lavoro, le quantità $\phi_{t+k|t}(j)$ e $\mathbb{P}(Y_{t+k}|\boldsymbol{y}^{(t)})$ ricavate nella (3.19) e nella (3.21).

In Tabella 3.5 riportiamo le previsioni puntuali per i giorni dal 22 dicembre 2023 al 31 dicembre 2023. Sebbene per alcuni valori lo scarto tra la temperatura realmente registrata a Vienna e quella prevista sia particolarmente elevato (ad esempio, per quello del 23 dicembre), il modello riesce, per lo meno, a cogliere l'andamento termico in termini di alternanza tra salita e discesa (esplicitato dalle frecce colorate nella Tabella 3.5). La radice dell'errore quadratico medio di previsione, che otteniamo come

$$RMSE = \sum_i (y_i - \hat{y}_i)^2,$$

risulta pari a 3.84.

3.2.2 Serie storica univariata, dati orari

Dal punto di vista della previsione puntuale, le prestazioni del modello hidden Markov applicato alle temperature giornaliere di Vienna lasciano a desiderare. Per vedere se, considerando i dati a granularità oraria al posto di quelli a granularità

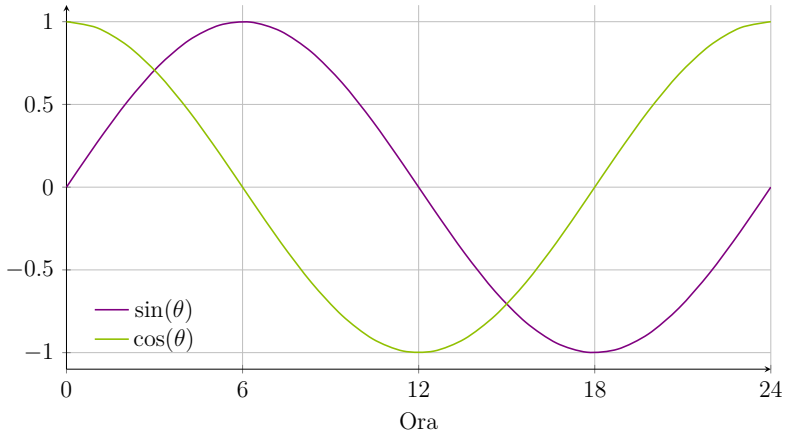


Figura 3.5: Trasformazione mediante funzioni trigonometriche dell'ora del giorno: seno e coseno di θ.

giornaliera, otteniamo scarti in valore assoluto più bassi tra temperatura reale e prevista, decidiamo di ripetere il lavoro effettuando delle previsioni orarie per le temperature nella città di Padova. Stavolta, nell'insieme di verifica inseriamo le 24 osservazioni relative al giorno di San Silvestro (31 dicembre) del 2023, mentre facciamo confluire le rilevazioni di tutti gli altri giorni dell'anno nell'insieme di stima.

Trasformazioni trigonometriche dell'ora

Dopo aver trasformato le variabili wind_direction e season in maniera analoga a quanto fatto per il modello (3.38), per tener conto dell'ora del giorno decidiamo di non effettuare un dummy coding. Quest'ultimo porterebbe alla creazione di ben $23 \times m$ parametri (aspetto rilevante in termini di costo computazionale) e non consentirebbe di tener conto del susseguirsi naturale delle ore; ad esempio, la "distanza" tra i livelli 6 e 7 sarebbe uguale a quella tra i livelli 6 e 22. Optiamo invece per l'uso di trasformazioni trigonometriche. L'ora h può essere espressa in radianti come

$$\theta = \frac{2\pi h}{24}, \quad h = 0, \ldots, 23;$$

il seno e il coseno di θ, insieme, riescono a rappresentare l'ora del giorno a guisa di un punto su un cerchio unitario, in un modo che cattura la natura ciclica della variabile, cosa apprezzabile dalla Figura 3.5. Da notare, ad esempio, come alle ore 23 siano assegnati valori di $\sin(\theta)$ e di $\cos(\theta)$ simili a quelli assegnati alle prime ore del giorno dopo.

Tabella 3.6: Coefficienti dell'HMM con 6 stati latenti applicato alla serie delle temperature orarie di Padova.

Coef.	Variabile	St.1	St.2	St.3	St.4	St.5	St.6
$\hat{\beta}_0$	Intercetta	31.11	6.98	28.22	18.15	21.92	32.08
$\hat{\boldsymbol{\beta}}_1$	season: estate	7.17	13.13	6.09	12.53	7.83	3.88
	season: inverno	−6.03	−3.00	−6.66	−4.34	−4.81	−9.49
	season: primavera	3.05	2.61	0.31	2.27	1.86	−0.96
$\hat{\beta}_2$	sin(ora)	−0.77	−1.43	−0.74	−1.28	−1.10	−0.85
$\hat{\beta}_3$	cos(ora)	−0.07	−0.70	−0.04	−0.36	−0.31	−0.10
$\hat{\beta}_4$	rain	0.07	0.15	−0.06	0.41	0.12	0.01
$\hat{\beta}_5$	wind_speed_10m	−0.05	−0.16	−0.09	−0.05	−0.05	−0.06
$\hat{\beta}_6$	relative_humidity_2m	−0.16	−0.05	−0.16	−0.13	−0.14	−0.15
$\hat{\beta}_7$	cloud_cover	$\simeq 0$	0.01	$\simeq 0$	0.01	0.01	$\simeq 0$
$\hat{\boldsymbol{\beta}}_8$	wind_direction: E/NE	−0.12	0.42	−0.03	0.17	0.09	−0.14
	wind_direction: E/SE	−0.06	0.06	$\simeq 0$	0.48	0.27	0.06
	wind_direction: S/SE	0.29	1.18	−0.04	0.12	0.20	0.01
	wind_direction: S/SW	0.02	0.99	−0.06	0.15	0.10	0.22
	wind_direction: W/SW	0.09	−0.01	−0.14	0.01	−0.08	0.22
	wind_direction: W/NW	−0.16	−0.79	−0.13	−0.11	−0.29	−0.13
	wind_direction: N/NW	−0.29	−0.22	0.03	−0.19	−0.23	−0.10
$\hat{\beta}_9$	wind_gusts_10m	$\simeq 0$	0.11	0.04	0.05	0.04	−0.01
$\hat{\beta}_{10}$	shortwave_radiation	$\simeq 0$	$\simeq 0$	$\simeq 0$	0.01	0.01	$\simeq 0$
$\hat{\beta}_{11}$	et0_fao_evapotranspiration	4.50	10.66	5.12	−8.06	−6.14	2.85

Modello e stime dei coefficienti

Il modello per la serie oraria delle temperature di Padova diventa

$$\begin{aligned}\mu = \beta_0 &+ \boldsymbol{\beta}_1 \cdot \mathsf{season} + \beta_2 \cdot \mathsf{sin(ora)} + \beta_3 \cdot \mathsf{cos(ora)} + \beta_4 \cdot \mathsf{rain} \\ &+ \beta_5 \cdot \mathsf{wind\_speed\_10m} + \beta_6 \cdot \mathsf{relative\_humidity\_2m} \\ &+ \beta_7 \cdot \mathsf{cloud\_cover} + \boldsymbol{\beta}_8 \cdot \mathsf{wind\_direction\_10m} + \beta_9 \cdot \mathsf{wind\_gusts\_10m} \\ &+ \beta_{10} \cdot \mathsf{shortwave\_radiation} + \beta_{11} \cdot \mathsf{et0\_fao\_evapotranspiration}.\end{aligned} \tag{3.39}$$

Effettuiamo una selezione del numero degli stati latenti del modello usando il BIC. Il valore di tale criterio per il modello con $m = 6$ risulta più basso rispetto a quello dei modelli con $m < 6$, ma più alto rispetto a quello del modello con $m = 7$. Al fine di favorire l'interpretabilità dell'analisi, e visto che la decrescita del BIC da $m = 6$ a $m = 7$ non è molto alta, consideriamo $m = 6$.

Le stime dei (numerosi) coefficienti sono consultabili in Tabella 3.6. Mettiamo in luce quelle relative alle variabili $\sin(\theta)$ e $\cos(\theta)$, per darne un'interpretazione e constatare il successo o l'insuccesso delle trasformazioni trigonometriche. Esami-

Tabella 3.7: Matrice delle probabilità di transizione dell'HMM applicato alla serie univariata delle temperature orarie di Padova.

	St.1	St.2	St.3	St.4	St.5	St.6
St.1	0.94	$\simeq 0$	0.01	$\simeq 0$	$\simeq 0$	0.05
St.2	$\simeq 0$	0.98	$\simeq 0$	0.02	$\simeq 0$	$\simeq 0$
St.3	0.01	$\simeq 0$	0.92	$\simeq 0$	0.03	0.03
St.4	$\simeq 0$	0.01	0.01	0.94	0.04	$\simeq 0$
St.5	$\simeq 0$	$\simeq 0$	0.03	0.04	0.93	$\simeq 0$
St.6	0.04	$\simeq 0$	0.03	$\simeq 0$	$\simeq 0$	0.93

nando contestualmente tutti gli stati latenti, le stime dei coefficienti associati alle suddette trasformazioni variano, rispettivamente, tra gli estremi $[-1.427, -0.743]$ e $[-0.698, -0.069]$: al diminuire di $\sin(\theta)$, cioè soprattutto quando l'ora è in un intorno di mezzogiorno, e al diminuire di $\cos(\theta)$, specialmente verso le diciotto, corrisponde un aumento delle temperature. Questo risultato è confortante e testimonia la bontà della trasformazione realizzata.

Sebbene siano più numerosi, come evinciamo dalla Tabella 3.7 gli stati latenti presentano una persistenza più alta rispetto a quelli individuati dal modello per la serie delle temperature di Vienna (Tabella 3.2). Molti γ_{ij}, infatti, sono stimati quasi a 0.

Performance previsiva

Utilizzando la matrice (3.7), costruiamo — ancora manualmente — le previsioni per le temperature del 31 dicembre 2023 nel capoluogo patavino. I risultati sono visibili in Tabella 3.8 e sono decisamente soddisfacenti. L'andamento delle temperature nelle loro discese e risalite è colto quasi alla perfezione, e gli scarti tra temperature osservate e attese sono molto più bassi rispetto a quelli ottenuti in Tabella 3.5. La radice dell'errore quadratico medio di previsione è circa pari a 0.27.

Abbiamo volutamente utilizzato covariate simili a quelle disponibili per i dati giornalieri per un confronto più equo tra le performance previsive dei modelli (3.38) e (3.39). I risultati incoraggianti riscontrati per la previsione di temperature orarie suggerirebbero di provare ad adattare un nuovo modello con tutte le variabili a disposizione. Bisognerebbe però abbinare a tale lavoro una strategia di selezione delle variabili, dal momento che, a priori, non è dato sapersi se tutte sono utili a spiegare la risposta. In questa dissertazione, preferiamo elaborare un nuovo modello sui dati giornalieri sfruttando la presenza, nel dataset, di serie storiche inerenti a più città.

Tabella 3.8: Performance previsiva del modello hidden Markov applicato alla serie storica delle temperature di Padova.

Ora	T. oss. (°C)	T. prev. (°C)	Ora	T. oss. (°C)	T. prev. (°C)
00:00	6.90	6.65	**12**:00	10.10 ↗	10.12 ↗
01:00	6.60 ↘	6.38 ↘	**13**:00	10.60 ↗	10.50 ↗
02:00	6.20 ↘	6.00 ↘	**14**:00	9.30 ↘	8.70 ↘
03:00	6.70 ↗	6.47 ↗	**15**:00	9.70 ↗	9.09 ↗
04:00	7.10 ↗	6.77 ↗	**16**:00	9.60 ↘	8.76 ↘
05:00	7.10 ↔	6.77 ↔	**17**:00	8.90 ↘	8.31 ↘
06:00	6.80 ↘	6.68 ↘	**18**:00	8.30 ↘	8.13 ↘
07:00	5.80 ↘	6.01 ↘	**19**:00	7.90 ↘	7.56 ↘
08:00	6.40 ↗	6.76 ↗	**20**:00	7.80 ↘	6.93 ↘
09:00	6.30 ↘	6.56 ↘	**21**:00	7.70 ↘	6.71 ↘
10.00	7.90 ↗	8.05 ↗	**22**:00	7.60 ↘	6.88 ↗
11.00	9.00 ↗	9.02 ↗	**23**:00	7.60 ↔	6.35 ↘

3.2.3 Serie storica multivariata, dati giornalieri

Applichiamo allora un modello hidden Markov a $m = 4$ stati latenti per la serie storica multivariata delle temperature massime registrate dal 2010 al 2023 per 223 città tra quelle elencate in Tabella 1.1 (escludendo Strasburgo e Alessandria poiché presentano dati mancanti). Utilizziamo le stesse variabili presenti nella formula del modello (3.38), con le medesime trasformazioni, senza inserire informazioni sulla collocazione spaziale delle città. Questo è giustificato dal fatto che, con il pacchetto `depmixS4`, purtroppo non è possibile inserire nella formula della distribuzione state-dependent degli effetti non lineari. D'altra parte, sembra insensato aggiungere lat e long come effetti lineari per una questione di orografia del territorio europeo: è vero che le temperature tendono a decrescere più o meno linearmente all'aumentare della latitudine, ma la presenza di catene montuose come Alpi, Pirenei e Alpi Dinariche fa sì che questa decrescita in realtà *non* sia lineare.

Le sequenze degli stati latenti sono assunte come indipendenti, mentre la stima dei parametri è unica per tutte le $K = 223$ serie componenti ("pooling parziale"). Suggeriamo di tornare al Paragrafo 3.1.4 per una descrizione più ampia dei possibili modi per incorporare serie longitudinali multivariate in un modello. In realtà, sebbene questa sia ovviamente una soluzione migliore rispetto all'ammettere un'unica sequenza di stati latenti per tutti i siti di rilevazione, avrebbe senso inserire esplicitamente nel modello un'informazione sulla vicinanza tra di essi, in modo da conferire a località vicine nello spazio sequenze simili. Ad esempio, potremmo far

Tabella 3.9: Matrice delle probabilità di transizione dell'HMM applicato alla serie multivariata delle temperature massime giornaliere.

	St.1	St.2	St.3	St.4
St.1	0.79	$\simeq 0$	0.16	0.05
St.2	0.02	0.85	0.12	$\simeq 0$
St.3	0.11	0.09	0.79	$\simeq 0$
St.4	0.12	$\simeq 0$	$\simeq 0$	0.88

dipendere la t.p.m. da un fattore che indichi l'appartenenza di una città ad un certo cluster ricavato tramite l'applicazione del k-means ai dati giornalieri. Avvisiamo che tale soluzione porterebbe ad un incremento notevole del tempo computazionale necessario per la stima.

Risultati

Anche senza incorporare informazioni sulla vicinanza tra città, la stima dei (numerosi) parametri del modello richiede un tempo considerevole: circa 25 minuti su un PC dotato di processore Intel Core i3-1115G4, con 8 GB di memoria RAM. Anche per questo motivo non effettuiamo una selezione del numero di stati latenti basata sul BIC; abbiamo verificato comunque, però, che il BIC del modello con $m = 4$ fosse inferiore a quello con $m = 2$ e $m = 3$.

In Tabella 3.9 riportiamo le probabilità di transizione tra i 4 stati latenti. Rispetto a quanto osservato in Tabella 3.2, notiamo una persistenza più elevata, specialmente per lo stato latente 4, dal quale è quasi impossibile transitare nel secondo e nel terzo stato. A giudicare dai coefficienti associati alle intercette (Tabella 3.10), lo stato 4 resta quello associato a condizioni climatiche più rigide. È confermato l'effetto positivo della variabile inerente all'anno, a testimonianza di quanto osservato durante le analisi esplorative. I coefficienti della variabile precipitation_hours non sono molto diversi da 0; purtroppo, il pacchetto `depmixS4` non fornisce la deviazione standard dei coefficienti. A meno di far partire la stima mediante algoritmo EM da punti di partenza diversi, e tener conto della variabilità dei coefficienti durante le replicazioni, non c'è un modo per dire se un coefficiente è statisticamente diverso o no da 0.

Performance previsiva

Valutiamo la performance previsiva del modello in maniera analoga a quanto fatto per la serie univariata: i risultati sono in Tabella 3.11. L'HMM applicato alla sola serie delle temperature di Vienna ha il pregio di ricalcare tutti gli aumenti e le

Tabella 3.10: Coefficienti dell'HMM con 4 stati latenti applicato alla serie storica multivariata delle temperature massime giornaliere.

Coef.	Variabile	St.1	St.2	St.3	St.4
$\hat{\beta}_0$	Intercetta	4.00	14.01	8.97	−5.38
$\hat{\beta}_1$	I(year−2010)	0.13	0.10	0.11	0.19
$\hat{\boldsymbol{\beta}}_2$	season: estate	0.74	0.13	0.59	0.03
	season: inverno	−3.81	−3.86	−3.92	−3.66
	season: primavera	−2.97	−3.39	−3.16	−3.01
$\hat{\beta}_3$	rain_sum	0.05	0.04	0.04	0.16
$\hat{\beta}_4$	precipitation_hours	0.02	−0.03	$\simeq 0$	$\simeq 0$
$\hat{\beta}_5$	wind_speed_10m_max	−0.08	−0.09	−0.08	−0.05
$\hat{\beta}_6$	wind_gusts_10m_max	−0.02	−0.01	−0.02	−0.02
$\hat{\boldsymbol{\beta}}_7$	wind_direction: E/NE	0.32	0.12	0.22	−0.01
	wind_direction: E/SE	0.92	0.28	0.67	0.48
	wind_direction: S/SE	1.49	0.54	1.09	0.97
	wind_direction: S/SW	1.58	0.44	1.02	1.87
	wind_direction: W/SW	1.03	0.24	0.63	1.47
	wind_direction: W/NW	0.41	$\simeq 0$	0.24	0.71
	wind_direction: N/NW	0.05	−0.09	0.02	−0.01
$\hat{\beta}_8$	shortwave_radiation_sum	−0.37	−0.67	−0.47	−0.46
$\hat{\beta}_9$	et0_fao_evapotranspiration	4.03	5.41	4.43	4.15
$\hat{\beta}_{10}$	snowfall_sum	−0.45	−0.18	−0.28	−0.23
$\hat{\beta}_{11}$	sun_hours	0.12	0.06	0.10	0.22
$\hat{\beta}_{12}$	sunshine_duration	0.62	0.42	0.51	1.04

diminuzioni di temperatura con riferimento al primo giorno dell'insieme di verifica; tuttavia, presenta alcuni scarti tra temperatura reale e prevista decisamente elevati (come quello del 23 dicembre). Il modello multivariato, invece, dal 29 dicembre al 30 dicembre prevede erroneamente un rialzo della temperatura. Tale rialzo è comunque quasi impercettibile: la qualità previsiva del modello multivariato sembra comunque essere migliore, cosa confermata dall'RMSE pari a 3.14, valore inferiore al 3.84 del modello univariato. Il miglioramento previsivo passando da serie storiche singole a serie multivariate non è prerogativa degli HMMs: la stessa cosa succede spesso nell'ambito dei modelli VAR (Vector Autoregressive Models).

Confronto tra sequenze di stati latenti

Come spiegato all'inizio della Sezione, nell'HMM multivariato non è stato incluso un effetto spaziale. Ci chiediamo se, in ogni caso, il modello sia in grado di definire una sequenza di stati latenti simile per località vicine nello spazio.

Tabella 3.11: Performance previsiva per Vienna del modello hidden Markov applicato alla serie multivariata delle temperature massime.

Giorno	Temperatura reale (°C)	Temperatura prevista (°C)	Scarto (°C)
22/12/2023	7.70	4.53	3.17
23/12/2023	7.10 ↘	4.04 ↘	3.06
24/12/2023	10.90 ↗	6.74 ↗	4.16
25/12/2023	13.60 ↗	12.59 ↗	1.01
26/12/2023	11.90 ↘	10.38 ↘	1.52
27/12/2023	10.10 ↘	6.51 ↘	3.59
28/12/2023	11.10 ↗	7.65 ↗	3.45
29/12/2023	13.50 ↗	8.84 ↗	4.66
30/12/2023	11.80 ↘	8.87 ↗	2.93
31/12/2023	4.30 ↘	6.23 ↘	-1.93

Il procedimento attuato per dare una risposta al quesito è il seguente. Per ciascuna delle 223 città, ricaviamo la rispettiva sequenza di stati latenti ottenuta con l'algoritmo di Viterbi. Successivamente, calcoliamo un semplice indice di somiglianza tra le sequenze relativa a Vienna e alla città c come

$$\frac{I(X_{i,Vienna} = X_{i,c})}{5103}, \quad i = 1, \ldots, 222. \tag{3.40}$$

Questo coincide col rapporto tra il numero di giorni contrassegnati dallo stesso stato latente per Vienna e c e la lunghezza totale delle serie storiche nell'insieme di stima. La Figura 3.6 consiste in un grafico di dispersione tra l'indice (3.40) e la distanza ortodromica in km tra Vienna e le altre città (la più breve che permette di congiungere due punti sulla Terra). Sono evidenziati in colore rosso i punti del grafico relativi alle altre città dell'Austria: Graz, Salisburgo, Villaco e Innsbruck. All'aumentare della distanza, diminuisce con un andamento pressoché quadratico la somiglianza tra le sequenze di stati latenti definita dall'indice (3.40). La correlazione di Spearman per le due variabili è pari a -0.64.

L'andamento meteorologico di Vienna è molto simile a quello di Bratislava, in Slovacchia, distante dalla capitale austriaca circa 55 km. Viceversa, com'è assolutamente realistico immaginare, un andamento degli stati latenti totalmente diverso da quello per Vienna si ha per la città marocchina di Marrakech. Dunque, possiamo affermare che, nonostante il modello sia agnostico nei confronti della posizione geografica delle città, riesca a riconoscere quali sono più vicine alle altre, assegnando loro una sequenza di stati latenti simili.

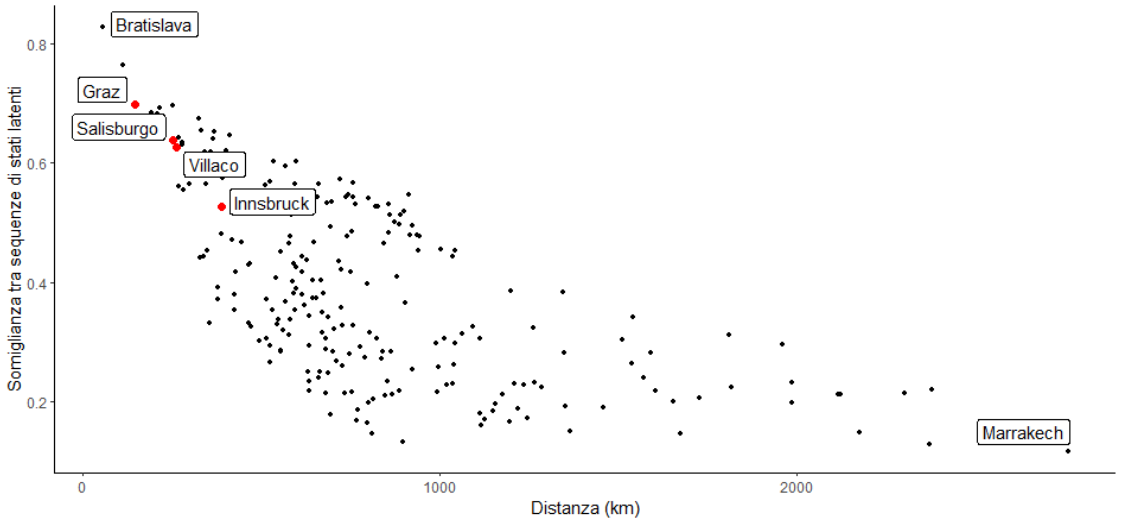


Figura 3.6: Somiglianza tra sequenze di stati latenti di Vienna e delle altre città in funzione della loro distanza ortodromica da Vienna.

3.2.4 Serie storica multivariata, dati orari

Per completezza, adattiamo un modello hidden Markov a $m = 6$ stati latenti sulla serie multivariata delle temperature orarie registrate nel corso del 2023 nei 54 comuni del Veneto della Figura 1.4. Omettiamo la tabella con i coefficienti, nella sostanza non molto diversi da quelli della Tabella 3.6 e comunque interpretabili con poca sicurezza da un punto di vista statistico, dato che non sono accompagnati dai loro errori standard. La matrice delle probabilità di transizione stimata è in Tabella 3.12 e risulta simile a quella di Tabella 3.7.

Nella Tabella 3.13, invece, mettiamo a confronto i valori reali della temperatura oraria registrata il 31 dicembre 2023 a Padova e i valori stimati dall'HMM per dati longitudinali. Il valore della metrica previsiva RSME è circa pari a 0.48, superiore allo 0.27 del modello per i soli dati di Padova. Differentemente da quanto osservato per i dati giornalieri, dunque, almeno in questo contesto il modello multivariato prevede in maniera meno accurata rispetto a quello univariato, anche se comunque le sue prestazioni previsive rimangono molto buone.

Il grafico della Figura 3.7 è concettualmente analogo a quello che troviamo in Figura 3.6: serve per capire il grado di somiglianza tra la sequenza di stati latenti individuata per Padova e quelle delle altre città. La correlazione di Spearman tra l'indice di somiglianza

$$\frac{I(X_{i,Padova} = X_{i,c})}{8736}, \quad i = 1, \ldots, 53, \tag{3.41}$$

Tabella 3.12: Matrice delle probabilità di transizione dell'HMM applicato alla serie multivariata delle temperature orarie.

	St.1	St.2	St.3	St.4	St.5	St.6
St.1	0.92	$\simeq 0$	0.04	$\simeq 0$	$\simeq 0$	0.04
St.2	$\simeq 0$	0.93	0.04	0.03	$\simeq 0$	$\simeq 0$
St.3	0.03	0.04	0.92	$\simeq 0$	$\simeq 0$	0.01
St.4	$\simeq 0$	0.03	$\simeq 0$	0.97	$\simeq 0$	$\simeq 0$
St.5	$\simeq 0$	$\simeq 0$	$\simeq 0$	$\simeq 0$	0.97	0.02
St.6	0.04	$\simeq 0$	0.01	$\simeq 0$	0.02	0.93

del tutto analogo a quello nella (3.40), e la distanza ortodromica tra Padova e un'altra località è negativa e pari a -0.70. Nel grafico, i puntini rossi indicano i comuni della provincia di Padova. Il comune veronese di Bosco Chiesanuova, situato a 1106 m di altitudine, è quello la cui sequenza di stati latenti risulta essere la più diversa da quella di Padova, anche se non è il più distante (quest'ultimo è Cortina d'Ampezzo). In conclusione, parimenti a quanto constatato in Figura 3.6, anche nel caso dell'HMM applicato ad una serie multivariata di dati orari la vicinanza spaziale viene ben modellata dalle corrispondenze tra sequenze di stati latenti, più o meno accentuate a seconda della distanza tra paesi.

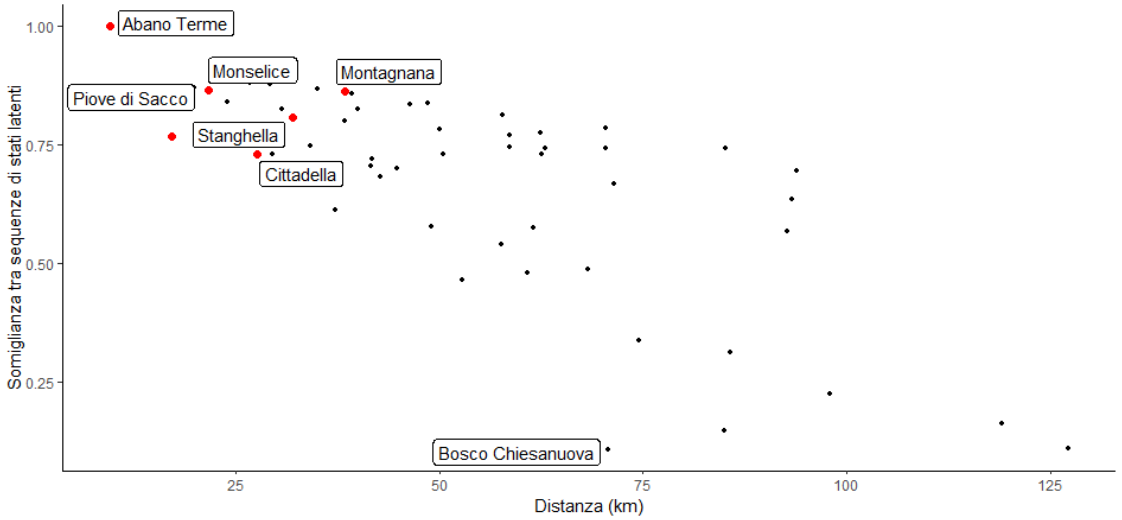


Figura 3.7: Somiglianza tra sequenze di stati latenti di Padova e degli altri comuni veneti in funzione della loro distanza ortodromica da Padova.

Tabella 3.13: Performance previsiva del modello hidden Markov applicato alla serie storica multivariata delle temperature in Veneto.

Ora	T. oss. (°C)	T. prev. (°C)	Ora	T. oss. (°C)	T. prev. (°C)
00:00	6.90	6.57	**12**:00	10.10 ↗	9.70 ↗
01:00	6.60 ↘	6.05 ↘	**13**:00	10.60 ↗	10.00 ↗
02:00	6.20 ↘	5.69 ↘	**14**:00	9.30 ↘	8.68 ↘
03:00	6.70 ↗	6.04 ↗	**15**:00	9.70 ↗	8.96 ↗
04:00	7.10 ↗	6.10 ↗	**16**:00	9.60 ↘	8.80 ↘
05:00	7.10 ↔	6.37 ↗	**17**:00	8.90 ↘	8.41 ↘
06:00	6.80 ↘	6.19 ↘	**18**:00	8.30 ↘	8.14 ↘
07:00	5.80 ↘	6.16 ↘	**19**:00	7.90 ↘	7.44 ↘
08:00	6.40 ↗	6.50 ↗	**20**:00	7.80 ↘	6.87 ↘
09:00	6.30 ↘	6.34 ↘	**21**:00	7.70 ↘	6.56 ↘
10.00	7.90 ↗	7.76 ↗	**22**:00	7.60 ↘	6.53 ↘
11.00	9.00 ↗	8.71 ↗	**23**:00	7.60 ↔	6.03 ↘

Capitolo 4

Altre tecniche per dati meteorologici

Nel Capitolo 3 abbiamo esplorato nel dettaglio, da un punto di vista sia teorico, sia applicativo, gli hidden Markov models. Il campo della modellazione di variabili atmosferiche è vasto e comprende una varietà di tecniche che possono offrire ulteriori spunti di approfondimento. Nel presente capitolo amplieremo la "cassetta degli attrezzi" per l'analisi dei dati meteorologici, toccando metodi di interpolazione, famiglie di distribuzioni e modelli per dati di rete.

4.1 Inverse distance weighting (IDW)

Uno dei modi concettualmente più intuitivi per stimare il valore di una variabile continua in presenza di dipendenza spaziale e temporale è l'inverse distance weighting (IDW). Questo metodo di interpolazione calcola la stima in un punto di interesse tramite una media ponderata dei dati, conferendo più peso alle osservazioni vicine nel tempo e nello spazio. Siano

$$\{Z(s_1,t_1), Z(s_2,t_1), \ldots, Z(s_{m_1},t_1), \ldots, Z(s_1,t_T), Z(s_2,t_T), \ldots, Z(s_{m_T},t_T)\}$$

i dati spazio-temporali, dove con $Z(s_i,t_t)$ indichiamo il valore osservato nel punto i

al tempo t. Il predittore IDW per la località s^* al tempo t^*, con $t_1 \leq t^* \leq t_T$, è

$$\hat{Z}(s^*, t^*) = \sum_{j=1}^{T} \sum_{i=1}^{m_j} w_{ij}(s^*, t^*) Z(s_i, t_j). \tag{4.1}$$

Definiamo nel seguente modo i pesi $w_{ij}(s^*, t^*)$:

$$w_{ij}(s^*, t^*) = \frac{\tilde{w}_{ij}(s^*, t^*)}{\sum_{k=1}^{T} \sum_{l=1}^{m_k} \tilde{w}_{lk}(s^*, t^*)}, \tag{4.2}$$

dove

$$\tilde{w}_{ij}(s^*, t^*) = k((s_i; t_j), (s^*, t^*); \theta). \tag{4.3}$$

La funzione $k((s_i; t_j), (s^*, t^*); \theta)$ viene chiamata "kernel function" e quantifica la vicinanza tra due osservazioni. Dipende da un parametro di regolazione (o "tuning") θ, che regola l'ampiezza del kernel stesso; all'aumentare di θ, cresce il numero di osservazioni che vengono mediate per il calcolo della grandezza meteorologica di interesse in (s^*, t^*). Esempi classici di funzione kernel sono la funzione radiale di base Gaussiana, che viene utilizzata anche nel contesto delle Support Vector Machine (SVM), e la seguente:

$$\tilde{w}_{ij}(s^*, t^*) = d((s_i, t_j), (s^*, t^*))^{-\theta}. \tag{4.4}$$

Nella (4.4) compare una semplice misura di distanza nel tempo e nello spazio tra (s_i, t_j) e (s^*, t^*), come può essere ad esempio quella Euclidea, elevata al coefficiente $-\theta$. Quest'ultimo può essere specificato a priori (un valore spesso usato nella pratica è $|\theta| = 2$), scelto tramite convalida incrociata, oppure — qualora non fosse ragionevole assumere una diminuzione costante del peso all'aumentare della distanza — stimato in maniera adattiva (Lu & Wong, 2008).

4.1.1 Interpolazione dei millimetri di pioggia

La tecnica dell'inverse distance weighting, dunque, stima il valore di una variabile in un certo punto dello spazio attribuendo opportuni pesi ai valori registrati in punti e tempi vicini — Equazione (4.1). Il metodo, molto simile al kriging, non è di previsione in senso stretto, perché il tempo t^* a cui fa riferimento il dato da stimare è compreso tra t_1 e t_T, primo e ultimo tempo osservato. L'IDW offre ugualmente un prezioso contributo in meteorologia quando l'intento è calcolare a posteriori una certa grandezza meteorologica per un'intera porzione di territorio, a partire dai dati "sparsi" derivanti dalle singole centraline. Pensiamo, ad esempio, al livello di umidità del suolo, parametro fondamentale per la pianificazione degli interventi di irrigazione e indice dello stato di un certo terreno in termini di siccità.

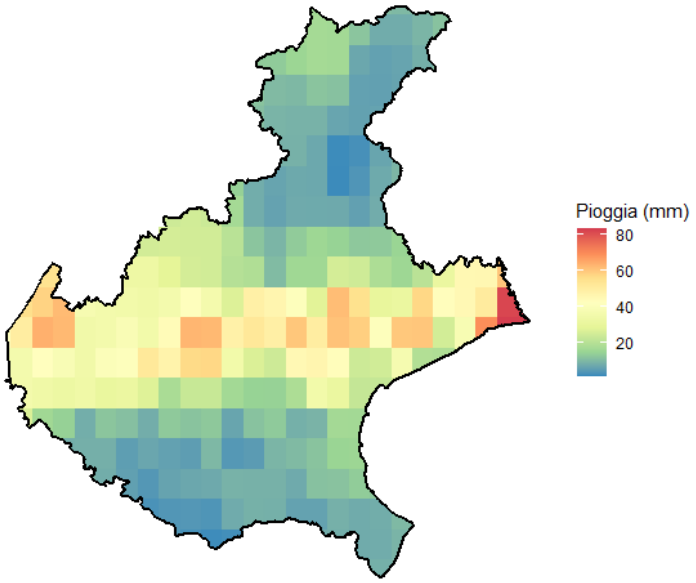


Figura 4.1: Interpolazione mediante IDW dei millimetri di pioggia caduti nel Veneto il 25 luglio 2023.

Per mostrare un esempio di applicazione dell'inverse distance weighting, interpoliamo su una griglia equamente spaziata i millimetri di pioggia caduti in Veneto il 25 luglio 2023 (giorno in cui a Bibione è stato rilevato il record riportato in Tabella 2.3). Onde ottenere un'interpolazione più precisa della misura presso i confini regionali, mettiamo assieme le informazioni provenienti dai due dataset a disposizione, aggiungendo alla somma giornaliera delle precipitazioni orarie per i 54 comuni del Veneto i dati giornalieri di città vicine, ma in un'altra regione (Ferrara, Mantova, Trento, Bolzano, Pordenone). Usiamo il kernel (4.4), fissando il parametro di tuning $|\theta|$ al valore 4.

I risultati dell'interpolazione sono apprezzabili in Figura 4.1, nella quale colori tendenti al rosso indicano una maggiore quantità di pioggia caduta al suolo. L'attività più intensa della linea temporalesca sembra aver attraversato il Veneto lungo un'asse che va dalla zona del Garda all'Adriatico. La distribuzione spaziale dei millimetri di pioggia caduti è piuttosto irregolare, come del resto accade spesso quando ci troviamo di fronte a temporali estivi piuttosto che a perturbazioni autunnali organizzate.

Sottolineiamo che, in questo specifico contesto, per il calcolo di $\hat{Z}(s^*, t^*)$ nei punti della griglia abbiamo sfruttato solo l'informazione sulla vicinanza spaziale e non su quella temporale. La procedura di inverse distance weighting potrebbe essere

migliorata, per esempio, regolando il parametro θ della (4.3) tramite convalida incrociata, oppure raccogliendo un numero ancora maggiore di osservazioni all'interno dell'area di interesse.

In generale — come fanno notare Stauffer et al. (2017) — sia kriging, sia inverse distance weighting appartengono alla famiglia delle tecniche di interpolazione esatta. Tali tecniche, sostanzialmente, fanno perno su un principio intuitivo e facile da capire come quello espresso dalla Prima legge della geografia di Tobler: « *everything is related to everything else, but near things are more related than distant things* ». Lo svantaggio è che non permettono di tener conto dei numerosi fattori locali che influenzano il meteo in una certa località: in presenza di covariate, è meglio adottare un approccio di tipo regressivo.

4.2 Due utili famiglie di distribuzioni

Presentiamo in questa sezione due famiglie distributive particolarmente adatte a modellare grandezze meteorologiche come la quantità di pioggia caduta e l'umidità relativa: la Tweedie e la Beta.

4.2.1 Modelli lineari generalizzati e distribuzione Tweedie

Sebbene sia particolarmente agevole sfruttare, per descrivere la distribuzione Tweedie, la teoria di base dei modelli lineari generalizzati, rimarchiamo che, in presenza di dati longitudinali relativi a più località nello spazio, è preferibile ricorrere a tecniche di modellazione più complesse — ad esempio, quelle che prevedono l'inserimento di un effetto casuale per tener conto della correlazione tra unità (GLMMs).

Modelli lineari generalizzati

La struttura di una famiglia di dispersione esponenziale univariata è

$$f_{Y_i}(y_i) = \exp\left\{\frac{y_i\gamma_i - b(\gamma_i)}{a_i(\varsigma)} - c(y_i, \varsigma)\right\}, \tag{4.5}$$

con γ_i parametro canonico (o naturale) di posizione e ς parametro di dispersione. Nei modelli lineari generalizzati (GLMs), la variabile risposta Y_i, $i = 1, \ldots, n$ appartiene a questa famiglia, e specificando la forma delle funzioni $a_i(\cdot)$, $b(\cdot)$ e $c(\cdot, \cdot)$ otteniamo un particolare modello parametrico (Salvan et al., 2020).

In ambito regressivo, la funzione che lega la media di y_i ad un predittore lineare η_i, $g(\cdot)$, può essere in linea di principio una qualsiasi funzione continua, monotona e differenziabile. Tuttavia, una scelta comune e conveniente è quella di utilizzare

come $g(\cdot)$ il link canonico, cioè quello tale per cui $\eta_i = \boldsymbol{x}_i^T\boldsymbol{\beta} = g(\mu_i)$. In questo caso, $\boldsymbol{X}^T\boldsymbol{y}$ è una statistica sufficiente per l'inferenza sui parametri $\boldsymbol{\beta}$.

Dalla (4.5) è immediato ricavare le espressioni per la log-verosimiglianza e il suo score:

$$l(\gamma_i, \varsigma; y_i) = \sum_{i=1}^{n} \frac{y_i\gamma_i - b(\gamma_i)}{a_i(\varsigma)} - \sum_{i=1}^{n} c(y_i, \varsigma), \tag{4.6}$$

$$\frac{\partial l(\gamma_i, \varsigma; y_i)}{\partial \gamma_i} = \frac{y_i - b'(\gamma_i)}{a_i(\varsigma)}. \tag{4.7}$$

A partire dalle due identità di Bartlett — valore atteso dello score pari a 0 e varianza dello score pari al valore atteso della sua derivata, cambiato di segno — possiamo ricavare le seguenti, importanti identità:

$$\mathsf{E}[Y_i] = b'(y_i) \tag{4.8}$$

$$\mathsf{Var}(Y_i) = a_i(\varsigma)\upsilon(\mu_i), \tag{4.9}$$

dove $\upsilon(\mu_i) = b''(\gamma_i)$ è la funzione di varianza, la quale descrive il modo in cui la varianza di Y_i dipende dalla propria media.

Millimetri di pioggia: la distribuzione Tweedie

Una variabile casuale Y_i ha distribuzione Tweedie con parametro di potenza p, $Y_i \sim \mathsf{TW}_p(\mu, \sigma^2)$, se

$$\mathsf{Var}(Y_i) = \sigma^2\upsilon(\mu_i) = \sigma^2\mu_i^p. \tag{4.10}$$

Puntualizziamo che, se $p = 0$, la distribuzione diventa una Normale; se $p = 1$ e $\sigma^2 = 1$, $\upsilon(\mu_i) = \mu_i$ e otteniamo una Poisson; quando $p = 2$, infine, la Tweedie coincide con la Gamma. Di particolare interesse, per lo meno nell'ambito meteorologico, sono i valori di p tra 1 e 2. In questo caso, la $\mathsf{TW}_p(\mu, \sigma^2)$ può essere rappresentata come la somma di un numero casuale N di Gamma indipendenti ed identicamente distribuite, con $N \sim \mathsf{Poi}(\lambda)$:

$$Y = \sum_{i=1}^{N} X_i, \qquad N \sim \mathsf{Poi}(\lambda), \qquad X_i \sim \Gamma(\alpha, \beta). \tag{4.11}$$

La caratteristica saliente delle distribuzioni Tweedie con potenza compresa tra 1 e 2 è di avere supporto nei reali non negativi, con la maggior parte della massa di probabilità situata in 0 (Dunn & Smyth, 2005). Pertanto, così come modelli ZIP (Zero-Inflated Poisson) e ZINB (Zero-Inflated Negative Binomial) sono d'aiuto nel gestire l'inflazione di zeri per dati discreti, il modello Tweedie può gestire dati continui con molti valori esattamente pari a 0. Le distribuzioni delle quantità orarie

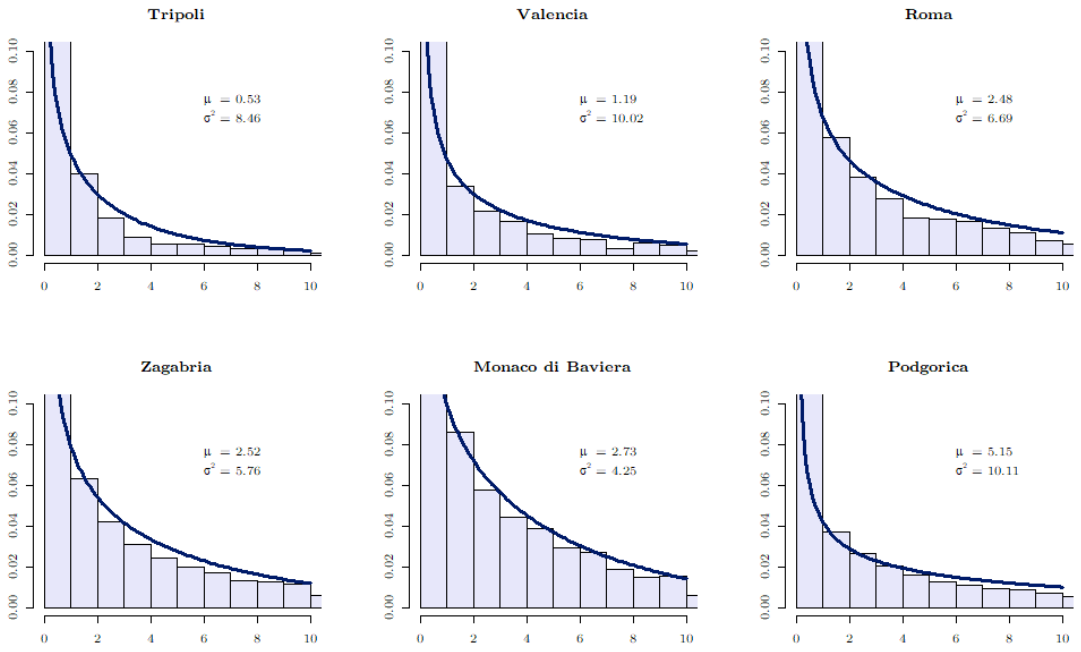


Figura 4.2: Istogrammi dei millimetri di pioggia giornalieri caduti in alcune città europee dal 2013 al 2020. In blu, la curva dei valori stimati da un GLM Tweedie con costante.

o giornaliere di pioggia, rilevate per un certo arco di tempo, tendono ad avere i connotati che le rendono adatte ad essere modellate tramite la distribuzione Tweedie (Figura 4.2).

4.2.2 Tasso di umidità e copertura nuvolosa: la distribuzione Beta

Per modellare valori continui con supporto limitato, come ad esempio quelli del tasso di umidità (altresì noto come umidità relativa), un possibile approccio è quello di applicare ai dati una trasformazione monotona che li mappi in $\mathbb{R}$ e utilizzare tecniche di regressione standard. Questo modo di procedere, tuttavia, presenta tre svantaggi (Cribari-Neto & Zeileis, 2010). In primo luogo, i parametri di regressione sono interpretabili in termini della media della trasformazione e non della variabile originale. In secondo luogo, tassi e proporzioni sono tipicamente eteroschedastici: mostrano una maggiore variazione intorno alla media e una minore variazione man mano che ci si avvicina ai limiti inferiore e superiore del supporto. Infine, l'asimmetria della distribuzione fa sì che le approssimazioni basate sulla Normale

per la stima degli intervalli e i test di ipotesi siano piuttosto imprecisi in campioni di piccole dimensioni.

La regressione Beta offre la possibilità di modellare valori appartenenti all'intervallo $(0, 1)$ senza ricorrere a trasformazioni, ed è eventualmente estendibile a variabili con generico intervallo di variazione (a, b). La distribuzione Beta, che per $\alpha, \beta > 0$ prende la forma

$$f_{Y_i}(y_i) = \frac{y_i^{\alpha-1}(1-y_i)^{\beta-1}}{B(\alpha,\beta)}, \quad B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}, \tag{4.12}$$

viene riparametrizzata in termini di media $\mu_i = \frac{\alpha}{\alpha+\beta}$ e precisione $\varsigma = \alpha + \beta$:

$$f_{Y_i}(y_i) = \frac{\Gamma(\varsigma)}{\Gamma(\mu_i\varsigma)\Gamma((1-\mu_i)\varsigma)} y_i^{\mu_i\varsigma-1}(1-y_i)^{(1-\mu_i)\varsigma-1}. \tag{4.13}$$

Il parametro ς, somma di α e β, viene chiamato "precisione" perché, per μ fissato, più alto è ς più bassa è la varianza di $\boldsymbol{Y}$. Per legare μ_i al predittore lineare $\boldsymbol{x}_i^T\boldsymbol{\beta}$ possiamo usare le funzioni logit, probit o complementary log-log. La distribuzione Beta è una famiglia esponenziale a due parametri, ma non appartiene propriamente alla classe che comprende, oltre alla Tweedie, le altre distribuzioni più comuni come la Normale, la Binomiale o la Poisson.

Nel campo della meteorologia, come anticipato, alcune variabili la cui distribuzione si avvicina ad una Beta sono quelle relative a umidità relativa (rapporto tra la quantità di vapore acqueo contenuto nell'aria e la quantità massima che l'aria sarebbe in grado di contenere ad una certa temperatura) e copertura nuvolosa. Un consiglio di lettura è l'articolo di Edilberto et al. (2019), che costruiscono un modello autoregressivo per l'umidità media settimanale misurata nella città di Rio Claro, nello stato di San Paolo (Brasile).

Quando ci troviamo nella situazione di dover fare una regressione Beta, occorre, come al solito, non applicare i modelli alla cieca, ma prestare attenzione ad un dettaglio. Se, utilizzando i dati a granularità oraria, adattiamo un modello di regressione Beta con sola intercetta considerando come risposta i valori di relative_humidity_2m rilevati in uno tra i 54 comuni veneti del dataset, la funzione `gam()` del pacchetto R `mgcv` restituisce un avviso di inaffidabilità delle stime. La causa è da imputare ad un eccessivo conteggio di valori di umidità relativa esattamente pari a 0 e ad 1. Consigliamo di vedere l'articolo di Liu & Kong (2015) per la discussione su come affrontare il problema dell'inflazione di 0 e 1 in un modello Beta (ZOIB, Zero-One-Inflated Beta).

4.3 Una rete di piogge

Prevedere il percorso di una cella temporalesca — specialmente quando i valori del CAPE (descritto al Paragrafo 2.2.2) o di altri indici di predisposizione atmosferica al maltempo sono elevati — è relativamente complicato anche disponendo di sofisticate tecnologie e di rilevazioni in tempo reale dai radar. È pur sempre vero che molte perturbazioni, specialmente nei mesi invernali, nel loro scorrere lungo un certo territorio non seguono direzioni completamente casuali. Prendendo come riferimento il Veneto, ad esempio, è più frequente constatare l'arrivo di perturbazioni da ovest o sud-ovest in moto verso levante, che non da nord-est verso ponente; è poi più facile che un temporale si sposti dalla montagna verso la pianura che non viceversa.

4.3.1 L'idea

A partire da queste premesse, si delinea l'opportunità di esplorare i pattern degli eventi precipitativi che hanno attraversato il Veneto nel 2023 facendo ricorso ad una rappresentazione di rete. In particolare, vogliamo costruire una rete composta dalle città del Veneto come nodi e con archi rappresentativi del numero di piogge o temporali che sono transitati da una città all'altra. La variabile `weather_code` è quella più utile allo scopo: può essere trasformata aggregando i livelli `61`, `63`, `65`, `71`, `73`, `75` nell'unico livello `1` ad indicare pioggia (o neve), contrapposto al livello `0` (assenza di pioggia) ottenuto dall'aggregazione dei livelli `0`, `1`, `2` e `3` (Tabella 1.6). L'idea che motiva questo lavoro è quella di arrivare a proporre alcuni modelli statistici per dati di rete volti ad investigare le dinamiche spaziali dei fenomeni atmosferici all'interno della regione, per mettere in luce percorsi e incidenza di piovaschi, rovesci e temporali nella zona.

La grande sfida posta da questo approccio è la costruzione della matrice di adiacenza pesata atta a tenere conto del numero di transiti di pioggia da nodo a nodo, cioè da comune a comune. Dai dati, è impossibile dedurre con certezza la traiettoria di una perturbazione. Idealmente, però, se in una certa fascia oraria di un giorno d'estate si registra temporale nella città i, e in quella successiva nella città j, — con i e j distinte tra loro e sufficientemente vicine — allora possiamo supporre, con un ragionevole grado di sicurezza, che una cella temporalesca sia transitata da i a j. Sintetizzando, l'assunzione che facciamo è la seguente (d_{ij} è la distanza tra località i e j in km):

$$\left.\begin{array}{l}\text{pioggia all'ora } t \text{ nella città } j\\ \text{pioggia all'ora } t-1 \text{ nella città } i\\ d_{ij} < s\end{array}\right\} \quad \Rightarrow \quad \text{transito di pioggia da } i \text{ a } j.$$

Algoritmo 1 Costruzione della matrice di adiacenza di perturbazioni.

1: **per** ogni ora t del 2023 a partire dalle 02:00 del 1° gennaio
2: **per** ogni comune i tra i 54
3: controlla: ha piovuto nel comune i all'ora t?
4: **se** sì, **allora**
5: ottieni l'insieme C_{t-1} dei comuni con pioggia all'ora $t-1$;
6: **se** $C_{t-1} \neq \emptyset$, **allora**
7: trova il comune j più vicino a i tra quelli in C_{t-1}
8: (se $j = i$, imposta j al secondo comune più vicino, se esiste);
9: **se** j è distante meno di una soglia s, **allora**
10: aggiungi 1 in posizione i, j alla matrice di adiacenza.

Nell'Algoritmo 1 estendiamo, mediante pseudocodice, il ragionamento espresso poc'anzi. Il punto di partenza è un dataset in formato largo, con un numero di righe pari al numero di comuni (54), e colonne di 0 e 1 derivanti dalla modifica dei valori orari di `weather_code` descritta in precedenza. Per ognuna delle colonne del suddetto dataset, iteriamo per comune alla ricerca del valore 1, che indica pioggia. Trovato un evento piovoso per il comune i all'ora t, cerchiamo i possibili comuni "mittenti" andando a perlustrare le condizioni atmosferiche nella fascia oraria $t-1$. Ad esempio, supponiamo di aver individuato pioggia alle 16:00 del 27 maggio per il comune di Monselice, e che alle 15:00 stesse piovendo a Pieve di Cadore, Abano Terme e Stanghella. Ottenuto l'elenco dei comuni papabili, C_{t-1}, per ognuno di essi calcoliamo la distanza in km da i e stabiliamo qual è il comune più vicino a i; nell'esempio, Stanghella. Infine, fissata una certa soglia s indicante la distanza massima ammissibile per lo spostamento realistico di una nube, controlliamo se la distanza tra il comune in C_{t-1} più vicino a i e i stesso è inferiore a s. In caso affermativo, incrementiamo di un'unità il conteggio nella posizione (i, j) della matrice di adiacenza. Stanghella e Monselice distano circa 11 km, quindi è plausibile che un fenomeno piovoso si sia spostato, nel tempo di un'ora, tra queste due città.

4.3.2 Problematiche

Il problema di fondo della procedura sta nel fatto che, a priori, non siamo in grado di stabilire l'entità della soglia s da usare nella riga `10` dell'Algoritmo 1. In altre parole, la difficoltà sta nel decidere qual è la distanza massima che possono coprire le nubi nel loro transito da una località all'altra. Alcuni temporali estivi possono essere piuttosto veloci, e raggiungere i 45/50 km/h, mentre in inverno le perturbazioni più organizzate tendono a produrre acquazzoni con movimento lento. Ma anche ammettendo di riuscire a trovare una soglia ottimale da un punto di vista fisico-

meteorologico, ci sono comunque situazioni che, con i dati a disposizione e con la metodologia presentata, è impossibile cogliere. Tornando all'esempio del paragrafo precedente, nulla garantisce che il temporale delle 16:00 a Monselice sia arrivato dalla più vicina Stanghella; potrebbe anche essere arrivato da Abano Terme, la cui distanza da Monselice è solo di poco superiore a quella che separa quest'ultima da Stanghella.

In aggiunta, utilizzando i dati orari non possiamo "separare" una perturbazione dall'altra e condizioni di maltempo persistente si traducono in un rapido aumento di conteggi nella matrice di adiacenza. Considerare i dati giornalieri al posto dei dati orari mitigherebbe questo aspetto, ma al contempo porterebbe alla formazione di legami opinabili da un punto di vista meteorologico, nel senso che potremmo affermare con poca sicurezza che

$$\left.\begin{array}{l}\text{pioggia al giorno } t \text{ nella città } j \\ \text{pioggia al giorno } t-1 \text{ nella città } i \\ d_{ij} < s\end{array}\right\} \quad \Rightarrow \quad \text{transito di pioggia da } i \text{ a } j.$$

Con la consapevolezza dei limiti intrinsechi alla metodologia sviluppata, procediamo ugualmente alla costruzione della matrice di adiacenza. La soglia s viene posta pari a 35 km. Nella Figura 4.3 mostriamo una rappresentazione grafica della rete di piogge in Veneto, per com'è stata descritta nei paragrafi precedenti. Valori più alti di s, ovviamente, porterebbero ad una rete più ricca di archi. Anziché usare algoritmi di force-directed placement per visualizzare la rete, posizioniamo ogni nodo in base alle sue coordinate geografiche. Il colore dei nodi rispecchia la loro betweenness: verde per i comuni periferici (più o meno tutti quelli ai confini della regione, con qualche eccezione), rosso per quelli che invece hanno un'alta centralità. Spessore e colore degli archi, invece, rispecchiano il peso attribuito all'arco a partire dalla matrice di adiacenza: molte connessioni si sono verificate tra nodi collegati da archi spessi e di colore blu.

Nonostante i (legittimi) dubbi sulla bontà delle assunzioni fatte per realizzare la rete, dall'immagine è possibile intravedere come l'asse principale delle perturbazioni sia stato quello ovest-est, dal momento che gli archi blu e spessi collegano per lo più nodi lungo questa direzione. Tali nodi sono situati prevalentemente nelle aree montane e pedemontane, altro aspetto perfettamente coerente con quanto ci si può attendere a partire da basilari conoscenze geografiche. Visivamente, possiamo poi individuare due cluster di rete, uno comprendente le città del Veneto sud-occidentale e uno con le città nord-orientali. Questo suggerisce la presenza di due "corridoi" preferenziali per il maltempo: uno dal basso Garda al rodigino, l'altro dalle Prealpi alla costa settentrionale. Portando la soglia s a 30 o 40 km, le considerazioni appena formulate non cambiano nella sostanza.

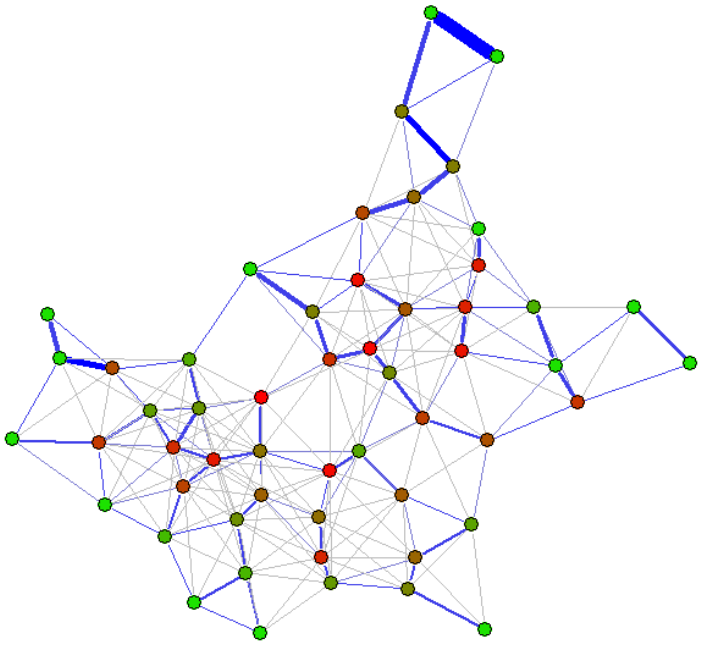


Figura 4.3: Rete dei transiti di pioggia da una città all'altra del Veneto, nell'anno 2023.

4.3.3 Il modello AME

I riscontri grafici di Figura 4.3 sono, nel complesso, in linea con le aspettative. Procediamo dunque con l'adattamento di un modello della classe AME (Additive and Multiplicative Effects). Utilizzati principalmente nell'ambito delle reti sociali, ma sufficientemente flessibili per trovare spazio in altri contesti, gli AME costituiscono un'estensione degli SRRM (Social Relations Regression Models). Questi ultimi permettono di misurare la relazione tra una certa variabile diadica (ossia relativa a due nodi) e altre variabili diadiche o di nodo, e hanno la forma

$$y_{ij} = \boldsymbol{\beta}^T \boldsymbol{x}_{ij} + a_i + b_j + \epsilon_{ij}, \tag{4.14}$$

dove y_{ij} è l'elemento in posizione (i, j) della matrice di adiacenza; $\boldsymbol{\beta}$ è un vettore di parametri associati alle covariate $\boldsymbol{x}_{ij}$; a_i e b_j sono effetti additivi di riga e di colonna; $\epsilon_{i,j}$ è un termine di errore Normale. La coppia di effetti (a_i, b_i) ha distribuzione Normale bivariata, per modellare la correlazione tra effetti di riga e di colonna relativi allo stesso nodo. Anche il vettore dei termini di errore ha una distribuzione Normale bivariata, con matrice di covarianze dipendente dai parametri σ^2 e ρ, che catturano variabilità addizionale e correlazione diadica.

Ad effetti additivi di riga e colonna, negli AME si aggiunge un effetto moltiplicativo $\boldsymbol{u}_i^T \boldsymbol{v}_j$, dove $\boldsymbol{u}_i$ e $\boldsymbol{v}_j$ sono vettori di fattori latenti relativi al comportamento dei nodi i e j come mittente e destinatario. Questo termine è in grado di quantificare la dipendenza triadica, cioè la tendenza della rete a formare triangoli. Nella matrice di adiacenza con i transiti di piogge da un comune veneto all'altro sono presenti molti zeri e relativamente pochi valori diversi da 0, alcuni dei quali, peraltro, molto grandi. In questi casi, la letteratura — rappresentata principalmente da Hoff (2018) — suggerisce, per mitigare la sovradispersione e in mancanza di alternative migliori, un approccio piuttosto drastico: modificare la risposta rendendola categoriale ordinale. La struttura del modello AME per risposte di questo tipo diventa la seguente:

$$\begin{aligned} y_{ij} &= \boldsymbol{\beta}^T \boldsymbol{x}_{i,j} + \boldsymbol{u}_i^T \boldsymbol{v}_j + a_i + b_j + \epsilon_{ij}, \\ s_{ij} &= g(y_{ij}). \end{aligned} \tag{4.15}$$

Ora y_{ij} è pensata come una variabile continua latente che regola il comportamento della variabile categoriale s_{ij}. La funzione che lega y_{ij} e s_{ij}, $g(\cdot)$, dev'essere non decrescente; il pacchetto R `ame` utilizza la funzione probit. Assumiamo inoltre $Cov(\boldsymbol{u}_i, \boldsymbol{v}_j) = \Psi$.

Inferenza nel modello AME e risultati

Quando la risposta è binaria o ordinale, come nel caso in esame, la stima dei parametri tramite la massimizzazione della verosimiglianza richiederebbe il calcolo di integrali difficilmente trattabili, derivanti dalla complessa struttura di dipendenza che stiamo tentando di modellare. Per questo motivo, l'inferenza sui parametri negli AME è di stampo Bayesiano e si avvale di un algoritmo MCMC (Markov Chain Monte Carlo), il Gibbs sampling, per simulare dalla distribuzione congiunta a posteriori dei parametri del modello. Come visibile dalla (4.15), questi ultimi comprendono gli effetti dei regressori sulla formazione di un legame tra un nodo e l'altro e quelli che governano le distribuzioni di riga, di colonna e dell'errore.

Stimiamo dunque per via Bayesiana i parametri di un modello AME sulla matrice di adiacenza degli scambi di pioggia, con elementi convertiti in quattro categorie (da "nessuno scambio" a "molti scambi"). I regressori utilizzati, scelti e modificati dopo numerosi tentativi, sono:

— lat.std, differenza tra latitudine del comune e latitudine del comune più a sud (Stienta);

— long.std, differenza tra longitudine del comune e longitudine del comune più a ovest (Peschiera del Garda);

Tabella 4.1: Medie e deviazioni standard a posteriori per il modello AME.

	Media	SD	z-value
lat.std.row	−5.251	1.091	−4.811
long.std.row	4.297	0.628	6.846
log(alt).row	1.633	0.222	7.339
lat.std.col	−6.900	1.515	−4.553
long.std.col	6.399	0.866	7.393
log(alt).col	2.117	0.306	6.914
log(dist)	−5.245	0.461	−11.374

— log(alt), logaritmo dell'altitudine in m del comune (trovata grazie alla funzione `geocode()` del pacchetto `openmeteo` e non presente nel dataset originale);

— log(dist), logaritmo della distanza in km tra comune i e j.

Ci aspettiamo, naturalmente, che quest'ultimo regressore diadico sia altamente significativo e con segno negativo, dal momento che, per costruzione, la rete non prevede legami tra nodi più distanti di una soglia s pari a 35 km.

I risultati della stima, cioè medie e deviazioni standard della distribuzione a posteriori, sono mostrati in Tabella 4.1. Come preannunciato, il coefficiente relativo alla distanza è significativo e con p-value vicino allo 0. Per quanto riguarda le componenti spaziali, dal modello emerge che la probabilità di osservare un legame, cioè un transito di pioggia, da una città all'altra si riduce all'aumentare della latitudine, cioè verso nord, e cresce all'aumentare della longitudine, cioè verso est. I comuni situati nella parte centro-meridionale del Veneto, dunque, hanno alta probabilità di essere nodi mittenti della pioggia, ma ancora più alta probabilità di essere i destinatari, perché i coefficienti di lat.std.col e long.std.col sono più alti dei corrispettivi effetti di riga.

Quest'ultima considerazione può anche risultare piuttosto sensata, ma purtroppo è evidente come il modello non sia in grado di modellare efficacemente la dinamica delle perturbazioni in Veneto. Infatti, nella parte settentrionale della regione abbiamo visto, in Figura 4.3, che in realtà ci sono state connessioni piuttosto intense, specialmente nel Bellunese. Ciò è confermato anche dai valore dei coefficienti di log(alt), positivi e significativamente diversi da 0. La difficoltà del modello di catturare questo aspetto è probabilmente da imputare a due aspetti: l'assenza di una griglia regolare di osservazioni, che bilanci lo scompenso numerico tra comuni nel nord del Veneto e nel resto del territorio, e la presenza di una soglia s a "pilotare" la formazione dei legami.

Conclusioni

I dataset ricavati da ERA5 sono stati una miniera di informazioni utili per comprendere l'andamento del clima e per mettere alla prova un vasto portfolio di strumenti adatti a descrivere, visualizzare, interpolare e prevedere dati di contesto meteorologico. Alcuni di questi strumenti, come l'Inverse Distance Weighting o i modelli hidden Markov, sono ben noti in letteratura; altri, invece, sono stati creati su misura per i dati a disposizione.

L'analisi esplorativa ha messo in luce soprattutto l'evidente aumento globale delle temperature occorso negli ultimi anni e la diminuzione delle precipitazioni a carattere nevoso sui principali rilievi montuosi. Le procedure di clustering, grazie a cui è stato possibile raggruppare località simili da un punto di vista climatico, sono state fatte con k-means e con l'ausilio di medie aritmetiche su periodi di tempo più o meno lunghi. Sarebbe interessante ripetere l'analisi sfruttando algoritmi che permettano, per loro natura, di tenere conto dell'ordinamento temporale delle osservazioni (come il "sequential k-means"). Lo studio dell'andamento della dissimilarità tra cluster, effettuato considerando i dati sulla concentrazione di alcuni inquinanti in Veneto, potrebbe inoltre essere esteso ad altre variabili oppure ripetuto con diversi indici di dissimilarità.

I modelli hidden Markov hanno dimostrato di essere strumenti potenti e versatili per modellare sequenze di temperature. In particolare, abbiamo osservato una performance previsiva piuttosto soddisfacente per la sequenza delle temperature orarie di Padova. Dataset ricchi di variabili come quelli creati offrono l'occasione di cimentarsi con molti altri tipi di HMM, ad esempio per risposte di conteggio, dicotomiche o addirittura multinomiali. Un'idea potrebbe essere quella di realizzare un HMM avente come risposta `weather_code`, al posto di trattarla come una variabile di stratificazione.

Al fine di ottenere, in generale, un adattamento ancora migliore ai dati (giornalieri o orari) da parte dei modelli hidden Markov, e auspicabilmente ancora migliori risultati previsivi, numerose sono le possibilità. L'inserimento di covariate a modellare la matrice delle probabilità di transizione potrebbe essere una di queste. La prassi comune è di utilizzare a tale scopo covariate variabili nel tempo. Tuttavia, avendo a che fare con dati localizzati anche nello spazio, potremmo prendere in considerazione lo spunto di modellare la matrice in questione in funzione della distanza tra località, facendo sì che località più vicine nello spazio abbiano sequenze di stati latenti simili (come sottolineato al Paragrafo 3.2.3). Va prestata cautela, però, al costo computazionale, che potrebbe lievitare e non di poco, soprattutto quando sono da modellare serie multivariate.

Un altro miglioramento può contemplare la sostituzione del dummy coding implementato per costruire la variabile `season` a partire dall'indicazione sul mese dell'anno con altre tecniche, più elaborate, per gestire la stagionalità nei modelli. In questa direzione, l'uso di funzioni trigonometriche, messo in pratica al Paragrafo 3.2.2, sembra dare buoni risultati, anche se l'interpretazione delle nuove variabili diventa leggermente macchinosa. Tali funzioni potrebbero essere applicate, inoltre, alla modellazione della direzione del vento, che nei dataset originali è espressa in gradi sessagesimali, tenendo presente che 0° indicano un vento di tramontana.

La modellazione degli spostamenti dei fenomeni atmosferici tramite dati di rete trova ben pochi riferimenti in letteratura; avvertiamo di non confondere quanto presentato nella dissertazione con le reti neurali, una tecnica di machine learning ben più utilizzata in ambito meteorologico. Il modello AME adattato alla trasformazione categoriale della matrice di adiacenza con i conteggi dei transiti di pioggia tra i 54 comuni veneti considerati non è riuscito a restituire output soddisfacenti. Tuttavia, con una raccolta o trasformazione dei dati fatta ad hoc per tenere traccia dei veri percorsi delle nubi, e con lo sviluppo di un algoritmo di Gibbs sampling adattato a dati di conteggio sovradispersi, non escludiamo di approdare a ben altre conclusioni. Per la rete di piogge in Veneto, un migliore approccio potrebbe partire da dati organizzati secondo una griglia rettangolare equispaziata di nodi (ad esempio, quella derivante delle rianalisi di ERA5 in formato GRIB). Comprendendo come nodi anche località in Trentino-Alto Adige, Emilia-Romagna e Friuli-Venezia Giulia, e ricavando da altre fonti sul web informazioni su data, ora e caratteristiche delle perturbazioni transitate nel territorio, aumenterebbe il grado di fiducia nella costruzione di un arco e i modelli potrebbero fornire risultati rilevanti.

Bibliografia

Bassi, F. & Ingrassia, S. (2022). *Statistica per analisi di mercato: metodi e strumenti*. Milano: Pearson.

Baum, L. E. & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The Annals of Mathematical Statistics*, 37(6), 1554–1563.

Ben Bouallègue, Z., Clare, M., Magnusson, L., Gascón, E., Maier-Gerber, M., Janoušek, M., Rodwell, M., Pinault, F., Dramsch, J., Lang, S., Raoult, B., Rabier, F., Chevallier, M., Sandu, I., Düben, P., Chantry, M., & Pappenberger, F. (2024). The rise of data-driven weather forecasting: A first statistical assessment of machine learning-based weather forecasts in an operational-like context. *Bulletin of the American Meteorological Society*.

Cappé, O., Moulines, E., & Rydén, T. (2005). *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer.

Casadei, M. & Finizio, M. (2024). Caldo, vento e piogge: fenomeni climatici sempre più estremi. *Il Sole 24 Ore*, (pp.2̃).

Copernicus Atmosphere Monitoring Service (2021). European air quality index calculation.

Cribari-Neto, F. & Zeileis, A. (2010). Beta regression in r. *Journal of Statistical Software*, 34, 1–24.

Dong, M., Yang, D., Kuang, Y., He, D., Erdal, S., & Kenski, D. (2009). Pm2.5 concentration prediction using hidden semi-markov model-based times series data mining. *Expert Systems with Applications*, 36(5), 9046–9055.

Dunn, P. K. & Smyth, G. K. (2005). Series evaluation of tweedie exponential dispersion model densities. *Statistics and Computing*, 15, 267–280.

Edilberto, C.-C., G., A. M., & Alberto, A. J. (2019). Beta meteorological time series: application to air humidity data.

Finlayson, B. L., Peel, M. C., & McMahon, T. A. (2007). Updated world map of the köppen-geiger climate classification. *Hydrology and Earth System Sciences*, 11(5), 1633–1644.

Grün, B. (2019). Model-based clustering. In *Handbook of mixture analysis* (pp. 157–192). Chapman and Hall/CRC.

Guttorp, P. & Zucchini, W. (1991). A hidden markov model for space-time precipitation. *Water Resources Research*, 27(8), 1917–1923.

Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., & Thépaut, J.-N. (2018). Era5 hourly data on single levels from 1940 to present.

Hersbach, H., de Rosnay, P., Bell, B., English, S., Buontempo, C., & Thepaut, J.-n. (2022). Status and plans of c3s earth system reanalysis following state-of-the-art data assimilation at ecmwf; era6 and beyond. In *AGU Fall Meeting Abstracts*, volume 2022 (pp. OS56B–02).

Hoff, P. D. (2018). Additive and multiplicative effects network models.

Langrock, R. & Zucchini, W. (2011). Hidden markov models with arbitrary state dwell-time distributions. *Computational Statistics & Data Analysis*, 55(1), 715–724.

Liu, F. & Kong, Y. (2015). Zoib: an r package for bayesian inference for beta regression and zero/one inflated beta regression.

Lu, G. Y. & Wong, D. W. (2008). An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences*, 34(9), 1044–1055.

Muñoz Sabater, J. (2019). Era5-land hourly data from 1950 to present. copernicus climate change service (c3s) climate data store (cds).

Salvan, A., Sartori, N., & Pace, L. (2020). *Modelli Lineari Generalizzati*. La Matematica per il 3+2, 124. Milano: Springer, 1st ed. 2020. edition.

Sansom, J. & Thomson, P. (2001). Fitting hidden semi-markov models to breakpoint rainfall data. *Journal of Applied Probability*, 38(A), 142–157.

Stauffer, R., Mayr, G. J., Messner, J. W., Umlauf, N., & Zeileis, A. (2017). Spatio-temporal precipitation climatology over complex terrain using a censored additive regression model. *International Journal of Climatology*, 37(7), 3264–3275.

Visser, I. & Speekenbrink, M. (2010). depmixs4: an r package for hidden markov models. *Journal of statistical Software*, 36, 1–21.

Wikle, C. K., Zammit-Mangion, A., & Cressie, N. (2019). *Spatio-temporal statistics with R*. Chapman and Hall/CRC.

Yu, S.-Z. (2010). Hidden semi-markov models. *Artificial Intelligence*, 174(2), 215–243. Special Review Issue.

Zippenfenig, P. (2024). Open-meteo.com weather api.

Zucchini, W., MacDonald, I. L., & Langrock, R. (2016). *Hidden Markov models for time series: an introduction using R*. Monographs on statistics and applied probability. Boca Raton [etc: CRC Press.

Appendice: codice R

In questa sezione vengono riportati i frammenti più importanti del codice utilizzato per estrarre i dati e svolgere le analisi descritte nei capitoli precedenti. Per eventuali chiarimenti o osservazioni, o per richiedere l'accesso a ulteriore codice, scrivere all'indirizzo filippounipd@gmail.com.

Codice per l'estrazione dei dati

La variabile relativa all'indice CAPE è stata costruita a partire dai dati in formato GRIB di ERA5 utilizzando Python. Per tutto il resto del lavoro di estrazione dei dati è stato adoperato il pacchetto R `openmeteo`. Si ricorda che l'API su cui si basa `openmeteo` presenta alcuni limiti di utilizzo (vedere 1.1.1).

Dati a frequenza giornaliera

Nel frammento di codice che segue si illustra come ricavare i dati meteorologici a frequenza giornaliera per la città di Barcellona, in Spagna. Allo stesso modo si possono estrarre i dati giornalieri di una qualsiasi altra città, avendo cura di specificarne il nome corretto o, in alternativa, le coordinate.

```
# Si aggiungono latitudine e longitudine ad ogni dataset
# usando le coordinate recuperate da
# https://geohack.toolforge.org/
# o mediante il comando openmeteo::geocode().

# install.packages('openmeteo')
library(openmeteo)
```

```
library(tidyverse)

v = weather_variables()
dhv = v[[4]] # daily_history_vars
cat(dhv, sep = ", ")
dhv = c(dhv, "rain_sum", "snowfall_sum", "sunshine_duration",
        "daylight_duration")
# Elenco delle variabili meteorologiche giornaliere da estrarre.

barcellona = weather_history(
  "Barcelona",
  "2010-01-01",
  "2023-12-31",
  daily = dhv
)
barcellona$city = "Barcellona"
barcellona$lat = 41.38879
barcellona$long = 2.15899
any(is.na(barcellona))
# Dati giornalieri per la citta' di Barcellona dal 2010 al 2023.
```

Codice 4.1: Esempio di estrazione dei dati giornalieri per la città di Barcellona.

Dati a frequenza oraria

Si presenta di seguito l'estrazione dei dati meteorologici a frequenza oraria per la città di Adria, in provincia di Rovigo. Come detto anche a riguardo dei dati giornalieri, si possono estrarre i dati orari di tutte le altre città semplicemente fornendo alle funzioni altri nomi o coordinate.

```
# install.packages('openmeteo')
library(openmeteo)
library(tidyverse)

v = weather_variables()
hhv = v[[3]] # hourly_history_vars
cat(hhv, sep = ", ")
# Elenco delle variabili meteorologiche orarie da estrarre.

poll = c("pm10", "pm2_5", "alder_pollen", "birch_pollen",
         "grass_pollen", "mugwort_pollen", "olive_pollen",
         "ragweed_pollen", "uv_index", "uv_index_clear_sky",
         "dust", "aerosol_optical_depth", "carbon_monoxide",
         "nitrogen_dioxide", "sulphur_dioxide", "ozone",
         "ammonia", "european_aqi")
# Elenco delle variabili sull'inquinamento da estrarre.

```

```
adria = weather_history(
  "Adria",
  "2023-01-01",
  "2023-12-31",
  hourly = hhv
)
adria$city = "Adria"
adria$lat = 45.05445
adria$long = 12.05599
any(is.na(adria))
apply(is.na(adria), 2, which)
adria$datetime = seq(as.POSIXct("2023-01-01 00:00:00"),
                     as.POSIXct("2023-12-31 23:00:00"),
                     by = "hour")
# Dati del 2023 per la citta' di Adria.
# Il NA deriva dal cambio dell'ora (da solare a legale).

adria.poll = air_quality(
  "Adria",
  "2023-01-01",
  "2023-12-31",
  hourly = poll
)
adria.full = cbind(adria, adria.poll[, -1])
# Inquinamento ad Adria.

write.csv(adria.full, file = "adria.csv", row.names = F)
# Dati orari per Adria.
```

Codice 4.2: Esempio di estrazione dei dati orari per un comune veneto.

CAPE

Si mostrano ora i comandi Python e R necessari alla conversione dei dati sull'indice CAPE dal formato GRIB al CSV, nonché all'associazione dei suddetti dati a quelli a granularità oraria per ognuna delle città considerate.

```
pip install ecmwflibs
pip install eccodes==1.3.1
pip install cfgrib
pip install xarray

import cfgrib
import pandas as pd
import xarray as xr

with xr.open_dataset('CAPE.grib', engine = 'cfgrib') as ds:
    df = ds.to_dataframe()
```

```

df.to_csv('cape-index.csv', index = True)
```

Codice 4.3: Estrazione dei dati sul CAPE (Convective Available Potential Energy).

```
# MODIFICA DEI DATI ------

cape <- read.csv(file.choose())
unique(cape$step)
unique(cape$surface)
unique(cape$number)
identical(cape$time,cape$valid_time)
cape$step = NULL
cape$number = NULL
cape$surface = NULL
cape$valid_time = NULL
cape$time = as.POSIXct(cape$time, format = "%Y-%m-%d %H:%M:%OS")
write.csv(cape, file = "cape-index.csv", row.names = F)


# ASSOCIAZIONE CAPE A DATI ORARI ------

city_coords = dati.orari %>%
  select(city, lat, long) %>% distinct()
# Coordinate di tutte le citta' presenti nel dataset.

cape_coords = cape %>%
  select(latitude, longitude) %>% distinct()
# Coordinate di tutte le citta' presenti nel dataset.

nearest_coords = data.frame(matrix(nrow = 54, ncol = 3))
columns = c("city", "n.lat", "n.long")
colnames(nearest_coords) = columns
# Dataframe che va a contenere le coordinate piu' vicine
# ai paesi.

for (i in 1:NROW(city_coords)) {
  nearest_coords[i, 1] = city_coords[i, 1]
  nearest_coords[i, 2] =
    cape_coords[which.min(abs(cape_coords$latitude - city_coords[i,
    2])), ]$latitude
  nearest_coords[i, 3] =
    cape_coords[which.min(abs(cape_coords$longitude - city_coords[i
   , 3])), ]$longitude
}
# Il dataset nearest_coords contiene le coordinate
# del dataset sul CAPE piu' vicine a quelle di ciascuno
# dei paesi del dataset con i dati orari.
```

```
42
43 city.names = sort(unique(dati.orari$city))
44 cbind(city.names, nearest_coords$city)
45
46 cape_list = list()
47 for (i in 1:54) {
48   cape_list[[i]] = cape %>%
49     filter(latitude == nearest_coords[i, 2],
50            longitude == nearest_coords[i, 3]) %>%
51     pull(cape)
52 }
53 names(cape_list) = city.names
54 # Creazione di una lista con i valori orari del CAPE
55 # per tutte le localita' presenti.
56
57 dati.orari$cape = unlist(cape_list)
58 # Aggiunta del CAPE.
```

Codice 4.4: Modifica dei dati sul CAPE e associazione ai dati orari.

Unione dei dataset

Al termine della raccolta dati, i dataset con le variabili meteorologiche sono stati uniti mediante il codice riportato in questo paragrafo.

```
1  filelist = list.files(pattern = "*.csv$")
2  # Dopo aver impostato come working directory quella
3  # contenente tutti i dataset relativi alle singole citta',
4  # si scrive la lista di tutti i file in formato .csv
5  # presenti nella cartella.
6
7  df_input_list <- lapply(filelist, read.csv)
8  names(df_input_list) <- gsub(filelist,
9                               pattern = "\\..*",
10                              replacement = "")
11 meteo <- dplyr::bind_rows(df_input_list,
12                           .id = "id")
13 write.csv(meteo, file = "meteo.csv",
14           row.names = F)
15 # Aggregazione nel dataset unico e scrittura.
```

Codice 4.5: Unione dei dataset con i dati meteorologici di tutte le città.

Codice per le analisi esplorative

Le analisi esplorative sono state condotte facendo ampio uso dei pacchetti R `tidyverse`, `ggplot2` e `ggrepel`.

Mappe

Si riporta il codice servito per realizzare, nell'ordine, le mappe geografiche delle Figure 1.3, 1.4, 2.7, 2.9. Riguardo alla mappa del Veneto, i file in formato GeoJSON necessari per la visualizzazione dei confini amministrativi sono stati esportati dal sito Overpass Turbo dopo un'opportuna query.

```
library(ggspatial)
library(rnaturalearth)
library(rnaturalearthdata)
world <- ne_countries(scale = "medium", returnclass = "sf")
# Pacchetti per la rappresentazione su mappa.

city_coords = full %>%
  select(city, lat, long) %>% distinct()
# Coordinate di tutte le citta' presenti nel dataset.

ggplot(data = world) +
  geom_sf(fill = "antiquewhite") +
  geom_point(data = city_coords,
             aes(x = long, y = lat), size = 1,
             shape = 21, fill = "darkblue") +
  coord_sf(xlim = c(-10, 29), ylim = c(30, 55), expand = FALSE) +
  theme(panel.background = element_rect(fill = 'skyblue'),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major = element_blank()) +
  annotation_north_arrow(location = "tr",
                         which_north = "true",
                         style = north_arrow_minimal) +
  xlab("") + ylab("")
```

Codice 4.6: Mappa con le città per le quali sono stati estratti i dati giornalieri.

```
paesi_coords = orari %>%
  select(city, lat, long) %>% distinct()
any(duplicated(paesi_coords$lat))
any(duplicated(paesi_coords$long))
# Coordinate di tutti i comuni presenti nel dataset.

library(sf)
veneto = st_read(file.choose())
veneto = veneto[1, ]
# Lettura di "veneto-etal.geojson".

library(ggrepel)
library(ggspatial)
```

```
ggplot(data = veneto) +
  geom_sf(fill = "antiquewhite") +
  geom_point(data = paesi_coords,
             aes(x = long, y = lat), size = 1,
             shape = 21, fill = "darkblue") +
  theme(panel.background = element_rect(fill = 'white'),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        panel.grid.major = element_blank(),
        plot.margin = margin(t = 1,
                             r = 1,
                             b = 1,
                             l = 1)) +
  geom_text_repel(data = paesi_coords,
                  aes(long, lat, label = city),
                  box.padding = 0.05,
                  point.padding = 0.05,
                  nudge_y = 0.025,
                  size = 4) +
  annotation_north_arrow(location = "tr",
                         which_north = "true",
                         style = north_arrow_minimal) +
  xlab("") + ylab("") +
  theme(panel.background = element_rect(fill = 'skyblue')) +
  coord_sf(xlim = c(min(paesi_coords$long) - 0.05, max(paesi_coords
   $long) + 0.1),
           ylim = c(min(paesi_coords$lat)  - 0.09, max(paesi_coords
   $lat) + 0.07))
```

Codice 4.7: Mappa con i comuni per i quali si hanno dati orari.

```
feb16 <- orari %>%
  filter(datetime == "2023-02-16 18:00:00")
# Dati alle 18 del 16 febbraio 2023.

pm10feb <- feb16 %>%
  select(city, lat, long, hourly_pm10)
# Polveri sottili PM10 il 16/02/2023 alle 18.

nomi = c("Abano Terme", "Adria", "Agordo", "Albaredo d'Adige",
         "Arzignano", "Asiago", "Badia Polesine",
         "Barbarano Vicentino", "Bassano", "Belluno",
         "Bibione", "Bosco Chiesanuova",
         "Castelfranco", "Castelmassa", "Cavarzere",
         "Cerea", "Chioggia", "Cittadella", "Conegliano",
         "Cortina d'Ampezzo", "Feltre", "Isola della Scala",
         "Jesolo", "Lonigo", "Malcesine", "Mel",
```

```
          "Mirano", "Monselice", "Montagnana",
          "Montebelluna", "Noventa Vicentina", "Oderzo",
          "Padova", "Peschiera del Garda",
          "Pieve di Cadore", "Piombino Dese",
          "Piove di Sacco", "Portogruaro", "Porto Tolle",
          "Possagno", "Rovigo", "San Dona' di Piave", "Soave",
          "Spiazzi", "Spresiano", "Stanghella",
          "Stienta", "Tregnago", "Treviso", "Valdagno",
          "Venezia", "Verona", "Vicenza", "Vittorio Veneto")
pm10feb$city = nomi
# Cambio di etichetta per fare il grafico.

library(ggrepel)
breaks <- c(0, 50, 100, 150, 200)
pm10feb$class <- cut(pm10feb$hourly_pm10,
                     breaks = breaks,
                     labels = FALSE)
ggplot(data = pm10feb) +
  geom_sf(data = veneto, fill = "antiquewhite", colour = "black") +
  geom_point(data = pm10feb,
             aes(x = long, y = lat, color = as.factor(class),
                 size = hourly_pm10)) +
  scale_size(guide = "none") +
  scale_color_manual(values = c("forestgreen", "darkgoldenrod1",
                                "darkred", "black"),
                     name = "Concentrazione PM10",
                     labels = c(...) +
  geom_text_repel(data = pm10feb %>%
                    filter(city %in% c("Cortina d'Ampezzo",
                                       "Verona")),
                  aes(x = long, y = lat, label = city),
                  size = 3,
                  box.padding = 0.5) +
  theme(panel.background = element_rect(fill = 'white'),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        legend.key = element_blank(),
        panel.grid.major = element_blank(),
        plot.margin = margin(t = 1,
                             r = 1,
                             b = 1,
                             l = 1)) +
  xlab("") + ylab("") +
  labs(fill = "PM10")
```

Codice 4.8: Concentrazione di PM10 alle 18 00 del 16 febbraio 2023 in Veneto.

```r
lug19 <- read.csv("lug19.csv")
# Dati sul CAPE in formato sf.

breaks = c(0, 2000, 3000, 4000, 5000, 6000, 6500)
ggplot(data = lug19) +
  coord_sf(xlim = c(min(lug19$longitude), max(lug19$longitude)),
           ylim = c(min(lug19$latitude), max(lug19$latitude)-0.5),
           expand = FALSE) +
  geom_contour_filled(data = lug19,
                      aes(x = longitude, y = latitude,
                          z = cape),
                      breaks = breaks) +
  scale_fill_brewer(palette = "YlGnBu") +
  geom_sf(data = veneto, fill = NA, colour = "black") +
  coord_sf(xlim = c(min(lug19$longitude), max(lug19$longitude)),
           ylim = c(min(lug19$latitude), max(lug19$latitude)-0.5),
           expand = FALSE) +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank()) +
  xlab("") + ylab("") +
  labs(fill = "CAPE")
```

Codice 4.9: Grafico del CAPE alle ore 18:00 del 19/07/2023.

Grafici

A titolo di esempio, si riportano i codici per realizzare i grafici delle Figure 2.1 e 2.8. Gli altri grafici presenti in questa tesi sono stati realizzati con tecniche analoghe.

```r
pioggia <- full %>%
  select(city, date, daily_precipitation_sum) %>%
  mutate(siccita = ifelse(daily_precipitation_sum <= 1, 1, 0))
# Dataframe con citta', data, precipitazioni giornaliere e una
# variabile dummy che indica se ha piovuto o no, dove con
# "pioggia" sono state intese precipitazioni piu' consistenti
# di 1 mm giornaliero (un litro d'acqua su un metro quadro di
# terreno).

pioggia <- pioggia %>%
  group_by(anno = lubridate::year(as.Date(date)),
           city = city,
           Group = cumsum(siccita != lag(siccita,
                                         default = first(siccita)))
  ) %>%
  mutate(consec_siccita = cumsum(siccita))
# Il dataframe "pioggia" contiene la colonna "consec_siccita" che
```

```r
# tiene il conto dei giorni consecutivi di siccita'.

max_consec_siccita <- pioggia %>%
  group_by(anno = lubridate::year(as.Date(date)), city) %>%
  summarise(max_consec_siccita = max(consec_siccita))
# Si costruisce un dataframe contenente il numero
# record per anno di giorni consecutivi senza pioggia.

ggplot(data = data.frame(max_consec_siccita %>%
       filter(city %in% c("Atene", "Madrid", "Nizza",
                          "Roma", "Tirana", "Zagabria"))),
       aes(x = anno, y = max_consec_siccita)) +
  geom_segment(aes(x = anno, xend = anno,
                   y = 0, yend = max_consec_siccita,
                   color = city),
               lwd = 1.25) +
  geom_point(size = 1.25) +
  theme_bw() +
  scale_x_continuous(breaks = seq(2010, 2023, by = 1),
                     labels = paste("'", seq(10, 23, 1),
                                    sep = "")) +
  scale_color_manual(values = c("#001489",
                                "#FFD100",
                                "#0055A4",
                                "#007A33",
                                "#DA291C",
                                "#C8102E")) +
  xlab("Anno") + ylab("Giorni consecutivi di siccita'") +
  facet_wrap(. ~ city, ncol = 3, scales = "free_x") +
  theme(axis.text.x = element_text(angle = 0),
        legend.position = "none",
        panel.grid = element_blank(),
        strip.background = element_rect(fill = "lightgoldenrod1"))
```

Codice 4.10: Periodi record di siccità in alcune città del bacino del Mediterraneo.

```r
aqi.city <- orari %>%
  group_by(id) %>%
  summarise(mean.aqi = mean(hourly_european_aqi))
# Indice complessivo di qualita' dell'aria per l'anno 2023,
# si vuole vedere quali sono stati i comuni con aria migliore
# e quelli con aria peggiore.

media.aqi = mean(aqi.city$mean.aqi)
# Media AQI per il Veneto (anno 2023).

dev.media = aqi.city %>%
  select(id, mean.aqi) %>%
  mutate(dev.aqi = mean.aqi - media.aqi) %>%
```

```
  select(-mean.aqi)
# Dataframe con deviazioni dalla media.

dev.media$id = nomi
ggplot(dev.media, aes(x = reorder(id, -dev.aqi), y = dev.aqi,
                      fill = dev.aqi)) +
  geom_bar(stat = "identity") + theme_bw() +
  theme(axis.text.x = element_text(angle = 0,
                                   vjust = 1,
                                   hjust = 1),
        axis.text.y = element_text(color = "black")) +
  scale_fill_gradient2(low = "green",
                       mid = "yellow",
                       high = "firebrick1") +
  xlab("") + ylab("Deviazione AQI dalla media regionale") +
  theme(legend.position = "none",
        panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.border = element_blank()) +
  geom_hline(aes(yintercept = 0),
             col = "black", lty = "dotted") + coord_flip()
# Le cinque citta' piu' virtuose sono Pieve di Cadore, Asiago,
# Feltre, Mel e Agordo. Le cinque citta' con l'aria peggiore sono
# Montebelluna, Mirano, Castelfranco, Piombino Dese e Padova.
```

Codice 4.11: Deviazioni dall'AQI per i 54 comuni veneti.

Clustering

Per il calcolo della dissimilarità settimanale tra cluster (Sezione 2.3) che dà luogo al grafico in basso a destra della Figura 2.13, ci si è serviti della funzione `calc.diss`.

```
pattern = character(52)
pattern[1:9] = c("01_", "02_", "03_", "04_", "05_", "06_",
                 "07_", "08_", "09_")
for (j in 10:52) {
  pattern[j] = paste(as.character(j), "_", sep = "")
}
# Pattern iniziale dei nomi di colonna da cercare (numero di
# ogni settimana da 01 a 52).

calc.diss <- function(data){

  means = numeric(52)
  # Vettore che conterra' le distanze medie tra i centroidi
  # dei cluster per le variabili sull'inquinamento
  # (indice di dissimilarita' tra cluster).

```

```
  for (i in 1:length(means)) {

    sett.i = data %>%
      select(city, grep(pattern = pattern[i], colnames(data)))
    sett.i = as.data.frame(sett.i) %>%
      mutate(across(-city, .fns = ~ as.numeric(scale(.x))))
    # i-esima settimana dell'anno, dati standardizzati.

    set.seed(44)
    km3 <- kmeans(sett.i[, -1], centers = 3, nstart = 20)
    # k-means con 3 cluster.

    means[i] = mean(dist(km3$centers))
    # Media delle distanze tra centroidi. Puo' essere considerata
    # come un indicatore di quanto dissimili sono i gruppi.
    # Piu' vicini sono tra loro i centroidi, piu' simili sono
    # i gruppi.

    print(i)
  }
  return(means)

}
```

Codice 4.12: Funzione per il calcolo della dissimilarità tra cluster prendendo in considerazione tutte le variabili sull'inquinamento.

Codice per HMM

Per l'adattamento dei modelli Hidden Markov è stato usato il pacchetto R `depmixS4`. Può essere utile, per confronto, provare a ripetere le stime mediante le funzioni di altri pacchetti come `HiddenMarkov` (dovrebbero fornire risultati molto simili).

HMM per una serie univariata

Di seguito, il codice per l'adattamento di un HMM alla serie storica univariata delle temperature massime giornaliere di Vienna (modello definito dall'Equazione (3.38)).

```
library(depmixS4)
set.seed(44)
mod <- depmix(daily_temperature_2m_max ~ ..., # covariate
                        nstates = ..., # numero stati
                        family = gaussian(),
                        data = train,
```

```
                          ntimes = 5103)
fmod <- fit(mod)
print(fmod)
```

Codice 4.13: HMM per la serie univariata delle temperature massime giornaliere di Vienna.

HMM per una serie multivariata

In presenza di una serie storica multivariata, basta agire sull'argomento `ntimes` e specificare il numero K di serie componenti.

```
library(depmixS4)
set.seed(44)
mod <- depmix(daily_temperature_2m_max ~ ..., # covariate
                          nstates = ..., # numero stati
                          family = gaussian(),
                          data = full %>% filter(!(day %in% c(22:31)
    & month == 12 & year == 2023)) %>%
                            filter(!(city %in% c("Alessandria", "
    Strasburgo"))),
                          ntimes = rep(5103, 223))
fmod <- fit(mod)
print(fmod)
```

Codice 4.14: HMM per la serie multivariata delle temperature massime giornaliere.

Come ricavare la matrice delle probabilità di transizione a partire da `fmod`? Ipotizzando di avere $m = 4$ stati latenti, è necessario agire nel modo esposto nel codice sottostante.

```
transition_mat <- rbind(getpars(getmodel(fmod, "transition", 1)),
                          getpars(getmodel(fmod, "transition", 2)),
                          getpars(getmodel(fmod, "transition", 3)),
                          getpars(getmodel(fmod, "transition", 4)))
colnames(transition_mat) <- c("1", "2", "3", "4")
```

Codice 4.15: Comandi per ricavare la matrice delle probabilità di transizione.