Inferenza Bayesiana del modello per blocchi massimi

Alessandro Quadrio

27 gennaio 2025

Sommario

In questo report viene affrontata la stima bayesiana per il modello dei blocchi massimi, con algoritmo Metropolis-Hastings; con un'applicazione alla stima delle temperature massime annuali registrate nel comune di Milano.

Indice

1 Introduzione all'Inferenza Bayesiana

L'approccio Bayesiano all'inferenza si basa su una concezione della probabilità intesa come "grado di credenza" di un osservatore razionale rispetto al verificarsi di un evento. Al contrario dell'approccio frequentista che interpreta la probabilità come la frequenza relativa di un evento osservato in un numero elevato di ripetizioni di un esperimento. Questa differenza che può apparire veniale, comporta una differenza decisiva nel processo inferenziale.

Nell'approccio frequentista, i parametri delle distribuzioni oggetto di inferenza sono considerati quantità fisse, non note e l'inferenza si basa esclusivamente sui dati osservati sperimentalmente. L'inferenza bayesiana invece tratta i parametri come variabili casuali, assegnando loro una distribuzione di probabilità iniziale e aggiornandola a seguito delle informazioni ricavate dall'esperimento.

Consideriamo un vettore $\mathbf{x}$ di n osservazioni indipendenti e identicamente distribuiti da $X \sim f(x|\theta)$ e supponiamo di voler effettuare inferenza sul parametro θ che caratterizza la distribuzione. Indichiamo con $\pi(\theta)$ la distribuzione a priori (prior) di θ. L'obiettivo è determinare la distribuzione a posteriori (posterior) di θ, $\pi(\theta|\mathbf{x})$, che può essere calcolata applicando il teorema di Bayes:

$$\pi(\theta|\mathbf{x}) = \frac{\pi(\theta)f(\mathbf{x}|\theta)}{\int_{\Theta} \pi(\theta)f(\mathbf{x}|\theta)\,d\theta} \tag{1}$$

Notiamo inoltre che il denominatore è una costante per θ possiamo scrivere

$$\pi(\theta|\mathbf{x}) \propto \pi(\theta)f(\mathbf{x}|\theta)$$

I principali vantaggi di questo approccio sono due. Innanzitutto, consente di integrare informazioni pregresse all'esperimento, arricchendo così l'analisi; inoltre, trattando il parametro come una variabile casuale, offre una gestione dell'incertezza più diretta e intuitiva.

1.1 Elicitazione della prior

L'aspetto più delicato dell'inferenza Bayesiana è senza dubbio l'elicitazione della prior, che aggiunge all'analisi un grado di soggettività. Ci sono vari metodi per fare ciò ma essi si dividono in due grandi gruppi:

- **Prior informative**, queste vengono scelte quando si hanno a disposizione robuste informazioni pre-sperimentali
- **Prior non informative**, che riflettono un'assenza di conoscenza preliminare, questo viene fatto utilizzando distribuzioni uniformi, oppure distribuzioni con varianze molto grandi.

C'è quindi un trade-off tra quante informazioni vengono aggiunte all'inferenza e il rischio di distorcere la stima aggiungendo informazione tendenziose.

1.2 Simulazione della posterior

Quando si considerano modelli statistici semplici un'opportuna scelta della forma funzionale della prior permette di rendere il calcolo della posterior molto semplice (attraverso l'equazione 1). Quando il modello diventa più complicato però questa possibilità viene meno e in molti casi il calcolo analitico della posterior è impossibile per via dell'integrale al denominatore. Bisogna quindi ricorrere a metodi numerici, principalmente vengono usati gli algoritmi Markov Chain MonteCarlo (MCMC). Questi metodi generano campioni dalla posterior creando dei processi markoviani la cui distribuzione stazionaria è la posterior (non nota).

Una volta che il processo raggiunge la distribuzione stazionaria si rimuovono le realizzazioni precedenti (burn in period) e le osservazione restanti θ_i vengono considerate come simulazioni

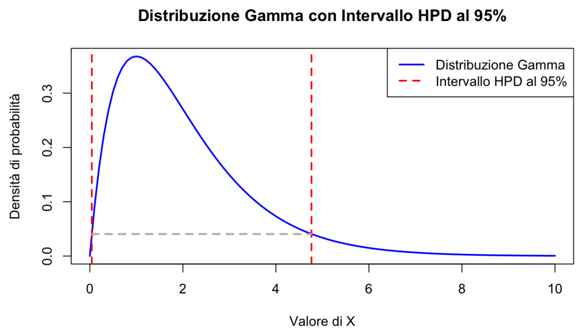

Figura 1: HPD di una Gamma (2,1)

della posterior. Un paio di esempi di questi algoritmi sono il Gibbs Sampling e il Metropolis-Hastings, quest'ultimo è quello che è stato implementato nella parte applicativa del report.

1.3 Intervalli di Credibilità

Il metodo più accurato per costruire un intervallo di credibilità è chiamato Highest Posterior Density (HPD).

Supponiamo di aver calcolato la posterior $\pi(\theta|\mathbf{x})$; per costruire un intervallo di credibilità HPD di livello $1-\alpha$, dobbiamo identificare i valori di θ più probabili della distribuzione a posteriori, in modo che costituiscano l' $(1-\alpha)\%$ della distribuzione.

Per calcolare gli estremi dell'intervallo non è possibile trovare una forma chiusa, ma si usano degli algoritmi numerici.

In figura 1 un esempio di un HPD di una distribuzione gamma.

1.4 Previsione

Supponiamo di voler prevedere un'osservazione futura x_{new} dato il dataset osservato $\mathbf{x}$. La distribuzione predittiva a posteriori (posterior predictive) si ottiene integrando il prodotto tra la distribuzione condizionale di x_{new} sui parametri θ e la distribuzione a posteriori di θ:

$$p(x_{\text{new}}|\mathbf{x}) = \int_{\Theta} f(x_{\text{new}}|\theta)\pi(\theta|\mathbf{x})\, d\theta \tag{2}$$

In pratica, quando la distribuzione a posteriori $\pi(\theta|\mathbf{x})$ non è disponibile in forma chiusa, questa integrazione viene approssimata utilizzando i campioni generati attraverso gli algoritmi MCMC. Per ciascuna estrazione $\theta^{(i)}$ dalla distribuzione a posteriori, si calcola un'osservazione predittiva simulata da $f(x_{\text{new}}|\theta^{(i)})$. Il risultato è una simulazione della posterior predictive.

2 Inferenza Bayesiana dei massimi

Verrà adesso trattato il modello dei blocchi massimi. Questo metodo prevede la suddivisione di una serie temporale in blocchi di uguale lunghezza (ad esempio, suddividendo dati giornalieri in blocchi annuali) selezionando il massimo valore osservato in ciascun blocco. Per semplicità considereremo il caso in cui i dati siano massimi annuali, ma la trattazione si può facilmente estendere al caso in cui il periodo di interesse sia un altro. L'analisi segue quanto riportato nel libro An Introduction to Statistical Modeling of Extreme Values di Stuart Coles [1].

2.1 Verosimiglianza

Secondo il teorema dei tre tipi, la distribuzione dei massimi, per blocchi di dimensione sufficientemente grande (in termini di osservazioni), converge a una distribuzione appartenente alla famiglia della Generalized Extreme Value (GEV), definita da:

$$F(x|\mu, \sigma, \xi) = \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\} \tag{3}$$

dove il dominio è $\left\{x : 1+\xi\frac{z-\mu}{\sigma} > 0\right\}$ e μ, σ e ξ sono rispettivamente parametri di posizione, scala e forma con μ e ξ definiti su $\mathbb{R}$ e σ definito su $\mathbb{R}^+$.

La funzione di densità della distribuzione GEV è definita come:

$$f(x|\mu, \sigma, \xi) = \frac{1}{\sigma}\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-1-\frac{1}{\xi}} \exp\left\{-\left[1+\xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\} \tag{4}$$

L'equazione 4 è la verosimiglianza del nostro modello, occorre adesso elicitare delle prior per i 3 parametri, una volta fatto ciò si potrà calcolare la posterior, o se ciò non fosse possibile, approssimarla numericamente.

2.2 Elicitazione della prior

Ponendo adesso $\phi = \log(\sigma)$ scriviamo la prior congiunta

$$\pi(\mu, \phi, \xi) = \pi(\mu)\pi(\phi)\pi(\xi)$$

La trasformazione del parametro σ viene effettuata per sfruttare il fatto che, essendo σ un valore positivo, il suo logaritmo può essere modellato con una distribuzione prior definita sull'intero insieme $\mathbb{R}$. Questo approccio semplifica l'elicitazione della prior e garantisce che i valori di σ generati siano sempre validi.

Le informazioni extra-sperimentatali nelle applicazioni ambientali dello studio dei massimi sono spesso scarse e quindi è comune utilizzare delle prior non informative. In particolare è ragionevole utilizzare come prior delle normali con varianza molto grande.

Inoltre è opportuno notare che, anche se nella prior le distribuzioni dei 3 parametri sono indipendenti questo non implica che ciò avvenga anche nella posterior.

2.3 Stima

In questo modello la posterior non è calcolabile analiticamente, è necessario utilizzare un metodo MCMC come il Gibbs Sampling o il Metropolis-Hastings algorithm.

Nell'analisi dei valori estremi, in particolare nell'ambito delle scienze ambientali, si è spesso interessati a stimare il livello di ritorno. Fissato un periodo m, il livello di ritorno x_m è quel livello che, in media, viene superato in un dato anno con probabilità $1/m$ o, in altre parole, che viene superato, in media, una volta ogni m anni.

Il livello di ritorno x_m può essere calcolato utilizzando la funzione quantile della distribuzione Generalized Extreme Value (GEV). Supponiamo di essere interessati al livello di ritorno con probabilità $p = 1/m$, dove m rappresenta il periodo di ritorno in anni.

Per la distribuzione GEV, il livello di ritorno z_p è definito come:

$$x_m = \begin{cases} \mu - \frac{\sigma}{\xi}\left[1 - \{-\log(1-m)\}^{-\xi}\right], & \text{per } \xi \neq 0, \\ \mu - \sigma\log\{-\log(1-m)\}, & \text{per } \xi = 0. \end{cases} \tag{5}$$

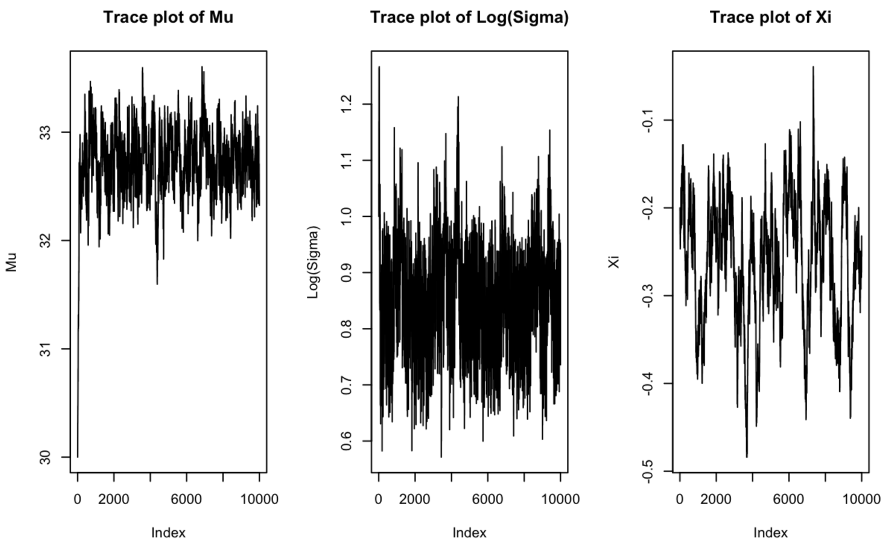

Figura 2: Trace plot delle catene markoviane dei 3 parametri

Se la posterior è già stata simulata, si può trovare una stima della distribuzione del livello di ritorno trasformando ogni vettore θ_i per l'equazione 5, ottenendo così delle simulazione dalla distribuzione a posteriori del livello di ritorno.

3 Caso Applicativo

Verrà adesso mostrata un'applicazione del modello alle temperature massime registrate nel comune di Milano. Il dataset utilizzato proviene dall'archivio storico di Meteoblue, che fornisce dati meteorologici per Milano dal 1940 ad oggi.

3.1 Modello con parametro di locazione costante

Iniziamo dal modello più semplice, in cui non consideriamo alcun tipo di dinamica temporale delle temperature massime.

Sono state utilizzate le seguenti prior:

- $\mu \sim \mathcal{N}(0, 100)$
- $\phi \sim \mathcal{N}(0, 100)$
- $\xi \sim \mathcal{N}(0, 10)$

3.1.1 Simulazione della posterior

La stima della posterior è stata fatta con un algoritmo Metropolis-Hastings con $10,000$ iterazioni, utilizzando come proposal un random walk con deviazioni standard 0.1, 0.1, 0.01 rispettivamente per μ, ϕ e ξ.

In figura 2 possiamo vedere che le catene sono arrivate a convergenza, questa non è perfetta per il parametro ξ ma comunque accettabile. Il tasso di accettazione è del 57%.

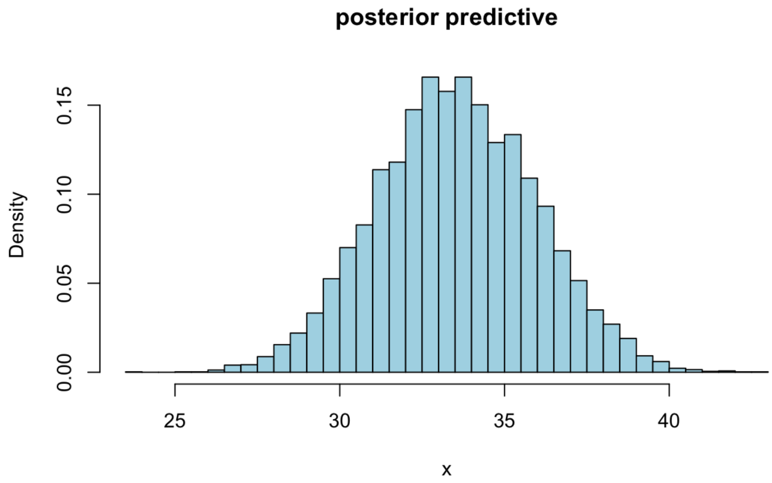

Figura 3: Approssimazione della posterior predictive con algoritmo Metropolis-Hastings.

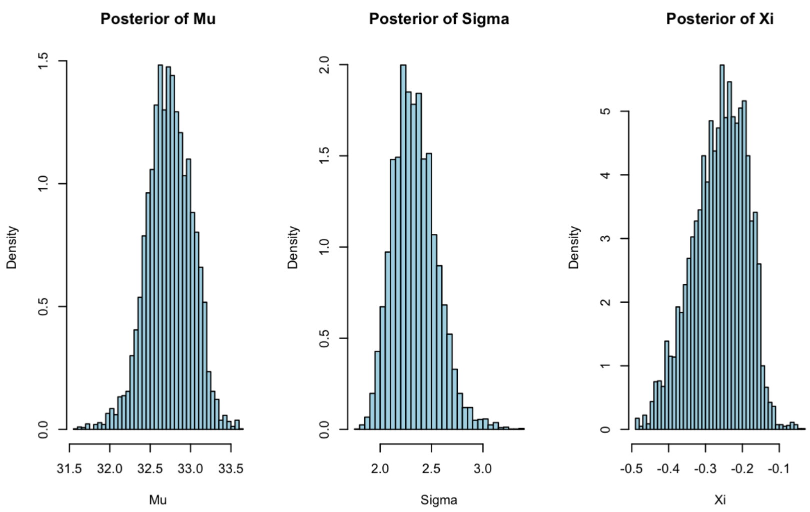

Figura 4: Approssimazione delle posterior dei parametri μ, σ e ξ con algoritmo Metropolis-Hastings

3.1.2 Posterior dei Parametri

In figura 4 vengono mostrate le approssimazioni delle posterior dei tre parametri: μ, σ e ξ.

I valori stimati per i parametri sono:

- μ: 32.74 (IC 95%: 32.22, 33.28)
- σ: 2.34 (IC 95%: 1.97, 2.76)
- ξ: -0.26 (IC 95%: -0.41, -0.14)

3.1.3 Posterior Predictive

La figura 3 mostra la posterior predictive per i massimi. La media della distribuzione predittiva è 33.60°C con un intervallo di credibilità al 95% compreso tra 29.08°C e 38.33°C.

I livelli di ritorno calcolati per diversi periodi sono:

- 10 anni: 36.71°C (IC 95%: 36.06, 37.43)
- 20 anni: 37.56°C (IC 95%: 36.83, 38.49)
- 50 anni: 38.45°C (IC 95%: 37.53, 39.63)
- 100 anni: 38.98°C (IC 95%: 37.88 40.34)

3.2 Modello con parametro di locazione con trend lineare

In questa sezione si esplora l'estensione del modello dei blocchi massimi per includere un trend lineare sul parametro di locazione μ, ovvero $\mu(t) = \beta_0 + \beta_1 t$, dove t rappresenta il tempo (in anni), e β_0 e β_1 sono i parametri da stimare. Questo approccio consente di catturare eventuali cambiamenti temporali nella media dei massimi annuali.

Sono state utilizzate le seguenti prior:

- $\beta_0 \sim \mathcal{N}(0, 100)$
- $\beta_1 \sim \mathcal{N}(0, 10)$
- $\phi \sim \mathcal{N}(0, 100)$
- $\xi \sim \mathcal{N}(0, 10)$

3.2.1 Simulazione della posterior

Per stimare i parametri del modello con il trend, è stato utilizzato un algoritmo di Metropolis-Hastings. L'algoritmo è stato eseguito per $10,000$ iterazioni, utilizzando come proposal un random walk con deviazioni standard 0.1, 0.01, 0.1, 0.01 rispettivamente per β_0, β_1, σ e ξ.

Le catene di Markov ottenute sono state analizzate per valutare la convergenza, come mostrato nei seguenti trace plot.

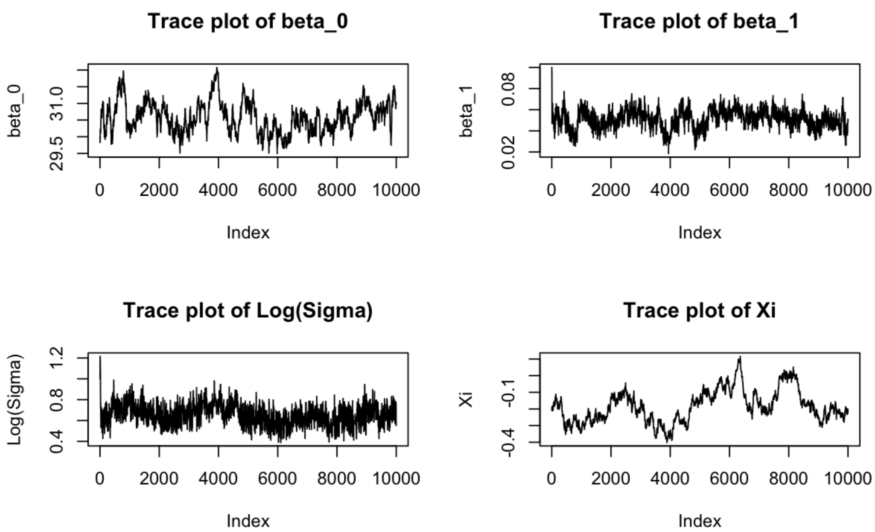

Figura 5: Trace plot delle catene Markoviane dei parametri β_0, β_1, $\log(\sigma)$ e ξ.

I trace plot mostrano un buon comportamento delle catene per i parametri β_0, β_1, $\log(\sigma)$ e ξ, anche se con qualche difficoltà di convergenza per il parametro di forma ξ. Il tasso di accettazione è stato del 37%.

3.2.2 Posteriori dei Parametri

Una volta ottenute le catene di Markov, sono stati calcolati i valori a posteriori dei parametri β_0, β_1, σ e ξ, dopo un periodo di burn-in di 2000 iterazioni. In figura 6, vengono mostrate le approssimazioni delle posterior dei parametri β_0, β_1, σ e ξ. I valori stimati sono:

- β_0: 30.62 (IC 95%: 29.63, 31.51)
- β_1: 0.0509 (IC 95%: 0.0.0327, 0.0686)
- σ: 1.95 (IC 95%: 1.59, 2.34)
- ξ: -0.20 (IC 95%: -0.394, 0.047)

Notiamo che, una volta introdotto il trend lineare, il parametro più influenzato è ξ che non è più significativamente negativo.

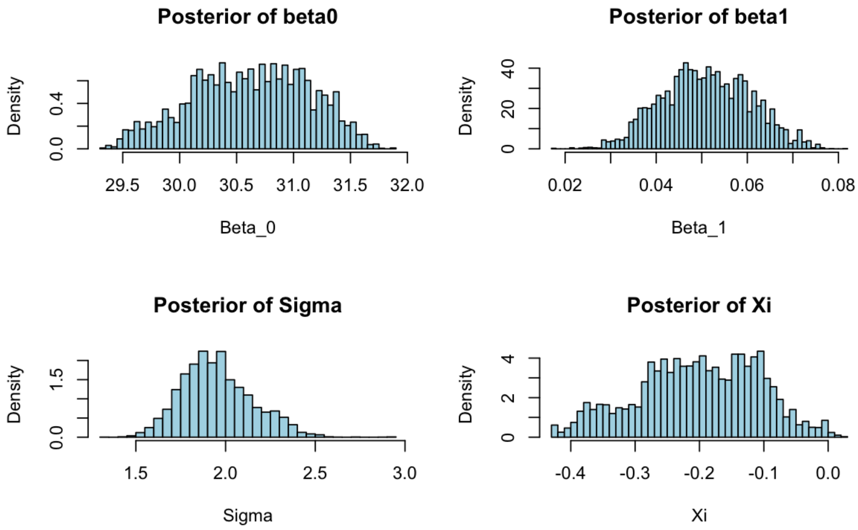

Figura 6: Approssimazione della posterior dei parametri β_0, β_1, σ e ξ con algoritmo Metropolis-Hastings.

3.2.3 Posterior Predictive

Successivamente, sono stati generati campioni dalla posterior predictive del massimo annuale di quest'anno (2025), fra 10, 50 e 100 anni, mostrati in figura 7. Gli intervalli di confidenza sono:

- 1 anni: 31.46 °C (IC 95%: 27.27, 35.44)
- 10 anni: 31.93 °C (IC 95%: 27.70, 35.89)
- 50 anni: 33.92 °C (IC 95%: 30.17, 38.08)
- 100 anni: 36.51 °C (IC 95%: 32.62, 40.94)

I livelli di ritorno in questo modello sono più complicati da calcolare, in quanto la distribuzione a posteriori è diversa nei diversi anni considerati. Il metodo che è stato scelto in questo report è il seguente: fissato il periodo di ritorno m sono state simulate le posterior predictive degli m anni successivi a quelli osservati. Considerando ognuna di queste serie storiche come una

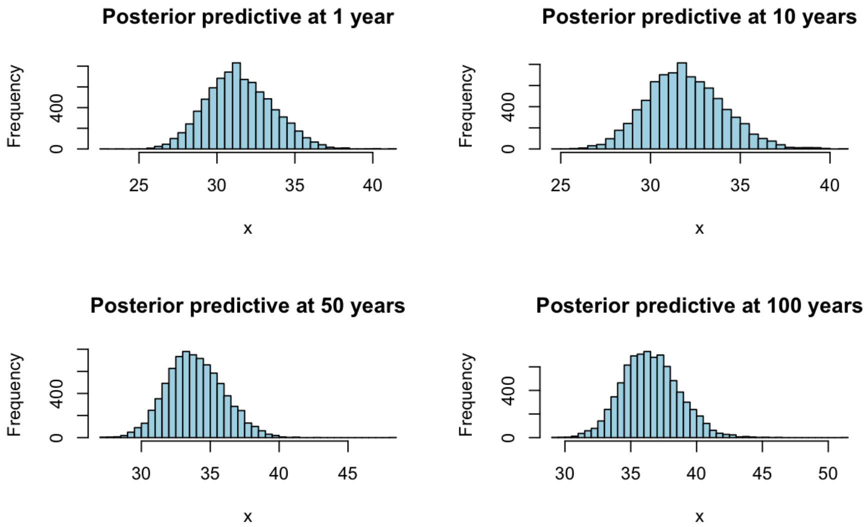

Figura 7: Approssimazione della posterior predictive con algoritmo Metropolis-Hastings a vari orizzonti temporali.

simulazione della distribuzione congiunta delle temperature massime negli m anni di interesse, è stato preso il quantile $\frac{m-1}{m}$ di ogni serie storica simulata come una simulazione del livello di ritorno a m anni. Di seguito i risultati:

- 10 anni: 33.86°C (IC 95%: 31.32, 37.36)
- 20 anni: 35.08°C (IC 95%: 32.23, 39.15)
- 50 anni: 37.08°C (IC 95%: 33.52, 41.87)
- 100 anni: 39.73°C (IC 95%: 35.19, 45.53)

4 Conclusioni

Il modello presenta dei vantaggi nella gestione dell'incertezza: l'utilizzo delle posterior predictive permette di creare intervalli di confidenza su una grande quantità di scenari futuri. Questo è un aspetto fondamentale nella previsione a un orizzonte temporale ampio. Gli svantaggi sono la difficoltà di definire delle prior informative e la necessità di utilizzare algoritmi simulativi che possono rivelarsi complessi da implementare.

Notiamo che, una volta aggiunto il trend lineare al parametro di locazione, anche le stime dei tempi di ritorno crescono linearmente. Questo suggerisce che, utilizzando una forma funzionale più flessibile, si potrebbe catturare un trend più complesso nei livelli di ritorno.

Un altro sviluppo futuro interessante potrebbe essere quello di introdurre dinamiche temporali anche sui parametri di scala e forma, per catturare dinamiche temporali più sofisticate.

A Implementazione del Codice

link per scaricare il dataset

```

library(evd)
library(coda)

# MODELLO CON PARAMETRO DI LOCAZIONE COSTANTE -----------------------------

set.seed(1)

Milano <- read.csv("Milano.csv")[,2:3]

# Definizione della funzione di verosimiglianza per la distribuzione GEV
likelihood <- function(params, data) {
  mu <- params[1]
  sigma <- exp(params[2])  # Trasformazione di sigma per assicurare che sia positivo
  xi <- params[3]

  # Verosimiglianza GEV (prodotto delle verosimiglianze individuali)
  n <- length(data)
  ll <- prod(dgev(data, loc = mu, scale = sigma, shape = xi))

  return(ll)
}

# Definizione della prior per i parametri
prior <- function(params) {
  # Prior normale per mu e xi
  mu_prior <- dnorm(params[1], mean = 0, sd = 100)
  phi_prior <- dnorm(params[2], mean = 0, sd = 100)
  xi_prior <- dnorm(params[3], mean = 0, sd = 10)

  return(mu_prior * phi_prior * xi_prior)
}

# Definizione dell'algoritmo di Metropolis-Hastings
metropolis_hastings <- function(data, init_params, n_iter, proposal_sd) {
  # Vettore iniziale dei parametri: mu, log(sigma), xi
  current_params <- init_params
  chain <- matrix(NA, nrow = n_iter, ncol = length(init_params))

  accept_ratio <- 0
  k <- 0
  l <- 0

  # Avvio di Metropolis-Hastings
  for (i in 1:n_iter) {
    # Proposta di nuovi parametri (usando una distribuzione normale)
    proposal <- current_params + rnorm(length(current_params), 0, proposal_sd)
    # Calcolo della verosimiglianza logaritmica per i parametri attuali e proposti
    num <- likelihood(proposal, data) * prior(proposal)
```

```r
    den <- likelihood(current_params, data) * prior(current_params)

    # Calcolo del rapporto di accettazione
    if(den==0) {accept_ratio <- 1; l <- l+1}
    else {accept_ratio <- num/den}

    # Accettazione o rifiuto della proposta basata sul rapporto di accettazione
    if (runif(1) < accept_ratio) {
      current_params <- proposal
      k <- k + 1
    }

    # Memorizzazione dei parametri attuali nella catena
    chain[i, ] <- current_params
  }

  return(list(chain, k,l))
}

# Inizializzazione dei parametri per mu, log(sigma), xi
init_params <- c(30,1,-.2)

# Esecuzione dell'algoritmo di Metropolis-Hastings
n_iter <- 10000
proposal_sd <- c(.1,.1,.01)  # Deviazione standard della distribuzione di proposta
chain <- metropolis_hastings(Milano[,2], init_params, n_iter, proposal_sd)

chain[[2]]/10000

par(mfrow = c(1, 3))
plot(chain[[1]][, 1], type = "l", main = "Trace plot of Mu", ylab = "Mu")
plot(chain[[1]][, 2], type = "l", main = "Trace plot of Log(Sigma)", ylab = "Log(Sigma)")
plot(chain[[1]][, 3], type = "l", main = "Trace plot of Xi", ylab = "Xi")

# Burn-in (scarto delle prime 1000 iterazioni)
burn_in <-2000
chain_burned <- chain[[1]][-(1:burn_in), ]

# Stime posteriori per i parametri
posterior_mu <- chain_burned[, 1]
posterior_sigma <- exp(chain_burned[, 2])  # Trasformazione inversa di sigma
posterior_xi <- chain_burned[, 3]

par(mfrow = c(1, 1))

post_pred <- numeric(8000)
for (i in 1:8000) {
  post_pred[i] <- rgev(1, loc = posterior_mu[i], scale = posterior_sigma[i],
                       shape = posterior_xi[i])
}
```

```
hist(post_pred, main = "posterior predictive", xlab = "x", col = "lightblue", border = "black", breaks = 50)

par(mfrow = c(1, 3))

# Mu
hist(posterior_mu, main = "Posterior of Mu", xlab = "Mu", col = "lightblue", border = "black", breaks = 50)

# Sigma
hist(posterior_sigma, main = "Posterior of Sigma", xlab = "Sigma", col = "lightblue", border = "black", breaks = 50)

# Xi
hist(posterior_xi, main = "Posterior of Xi", xlab = "Xi", col = "lightblue", border = "black", breaks = 50)

# Stime puntuali

(mean(post_pred))
(mean(posterior_mu))
(mean(posterior_sigma))
(mean(posterior_xi))

# IC
# Calcola l'HPD con livello di confidenza del 95%

post_pred_mcmc <- as.mcmc(post_pred)
print("intervallo di confidenza al 95% per la posterior predictive")
HPDinterval(post_pred_mcmc, prob = 0.95)

mu_mcmc <- as.mcmc(posterior_mu)
print("intervallo di confidenza al 95% per mu")
HPDinterval(mu_mcmc, prob = 0.95)

sigma_mcmc <- as.mcmc(posterior_sigma)
print("intervallo di confidenza al 95% per sigma")
HPDinterval(sigma_mcmc, prob = 0.95)

xi_mcmc <- as.mcmc(posterior_xi)
print("intervallo di confidenza al 95% per xi")
HPDinterval(xi_mcmc, prob = 0.95)

# Livello di ritorno

return_periods <- c(10, 20, 50, 100)  # Periodi di ritorno in anni
return_levels <- sapply(return_periods, function(m) {
  mu <- mean(posterior_mu)
  sigma <- mean(posterior_sigma)
  xi <- mean(posterior_xi)
  mu - sigma / xi * (1 - (-log(1 - 1/m))^(-xi))
})
return_levels

for (i in 1:4) {
  t <- return_periods[i]
  rl_ic <- posterior_mu - posterior_sigma / posterior_xi * (1 - (-log(1
```

```r
      - 1/t))^(-posterior_xi))
  rl_ic <- sort(rl_ic)
  print(paste("Intervallo di confidenza al", t, "%"))
  print(HPDinterval(as.mcmc(rl_ic), prob = 0.95))
}

# MODELLO CON TREND LINEARE ----------------------------------------------


set.seed(1)
# Definizione della funzione di verosimiglianza con trend lineare nel parametro di localizzazione
likelihood_trend <- function(params, data, time) {
  beta0 <- params[1]
  beta1 <- params[2]
  sigma <- exp(params[3])  # Trasformazione di sigma per assicurare che sia positivo
  xi <- params[4]

  # Parametro di localizzazione dipendente dal tempo
  mu_t <- beta0 + beta1 * time

  # Calcolo della verosimiglianza GEV
  ll <- prod(dgev(data, loc = mu_t, scale = sigma, shape = xi))

  return(ll)
}

# Definizione della prior con parametro aggiuntivo beta1
prior_trend <- function(params) {
  beta0_prior <- dnorm(params[1], mean = 0, sd = 100)
  beta1_prior <- dnorm(params[2], mean = 0, sd = 10)
  phi_prior <- dnorm(params[3], mean = 0, sd = 100)
  xi_prior <- dnorm(params[4], mean = 0, sd = 10)

  return(beta0_prior * beta1_prior * phi_prior * xi_prior)
}

# Modifica dell'algoritmo di Metropolis-Hastings
metropolis_hastings_trend <- function(data, time, init_params, n_iter, proposal_sd) {
  current_params <- init_params
  chain <- matrix(NA, nrow = n_iter, ncol = length(init_params))

  accept_ratio <- 0
  k <- 0
  l <- 0

  for (i in 1:n_iter) {
    proposal <- current_params + rnorm(length(current_params), 0, proposal_sd)

    # Calcolo della verosimiglianza logaritmica per i parametri attuali e proposti
    num <- likelihood_trend(proposal, data, time) * prior_trend(proposal)
    den <- likelihood_trend(current_params, data, time) * prior_trend(
```

```r
        current_params)

    if (den == 0) {
      accept_ratio <- 1
      l <- l + 1
    } else {
      accept_ratio <- num / den
    }

    # Accettazione o rifiuto della proposta
    if (runif(1) < accept_ratio) {
      current_params <- proposal
      k <- k + 1
    }

    chain[i, ] <- current_params
  }

  return(list(chain, k, l))
}

# Preparazione della variabile temporale
time <- seq_along(Milano[, 2])

# Inizializzazione dei parametri per beta0, beta1, log(sigma), xi
init_params <- c(30, 0.1, 1, -0.2)

# Esecuzione dell'algoritmo di Metropolis-Hastings
n_iter <- 10000
proposal_sd <- c(0.1, 0.01, 0.1, 0.01)
chain <- metropolis_hastings_trend(Milano[, 2], time, init_params, n_iter, proposal_sd)

chain[[2]]/10000

par(mfrow = c(2, 2))
plot(chain[[1]][, 1], type = "l", main = "Trace plot of beta_0", ylab = "beta_0")
plot(chain[[1]][, 2], type = "l", main = "Trace plot of beta_1", ylab = "beta_1")
plot(chain[[1]][, 3], type = "l", main = "Trace plot of Log(Sigma)", ylab = "Log(Sigma)")
plot(chain[[1]][, 4], type = "l", main = "Trace plot of Xi", ylab = "Xi")

# Burn-in (scarto delle prime 1000 iterazioni)
burn_in <-2000
chain_burned <- chain[[1]][-(1:burn_in), ]

# Stime posteriori per i parametri in osservazioni future

posterior_beta0 <- chain_burned[, 1]
posterior_beta1 <- chain_burned[, 2]
posterior_sigma <- exp(chain_burned[, 3])  # Trasformazione inversa di sigma
posterior_xi <- chain_burned[, 4]

par(mfrow = c(2, 2))
```

```r
for (t in c(1,10,50,100)){
  post_pred <- numeric(8000)
  for (i in 1:8000) {
    post_pred[i] <- rgev(1, loc = posterior_beta0[i] + posterior_beta1[i] * t,
                         scale = posterior_sigma[i],
                         shape = posterior_xi[i])
  }

  hist(post_pred, main = "posterior predictive", xlab = "x", col = "lightblue", border = "black", breaks = 50)
  post_pred <- sort(post_pred)

  print(mean(post_pred))

  print(paste("Intervallo di confidenza a", t, "anni"))
  print(HPDinterval(as.mcmc(post_pred), 0.95))
}
post_pred <- sort(post_pred)
par(mfrow = c(2, 2))

# Beta_0
hist(posterior_beta0, main = "Posterior of beta0", xlab = "Beta_0", col = "lightblue", border = "black", breaks = 50)

hist(posterior_beta1, main = "Posterior of beta1", xlab = "Beta_1", col = "lightblue", border = "black", breaks = 50)

# Sigma
hist(posterior_sigma, main = "Posterior of Sigma", xlab = "Sigma", col = "lightblue", border = "black", breaks = 50)

# Xi
hist(posterior_xi, main = "Posterior of Xi", xlab = "Xi", col = "lightblue", border = "black", breaks = 50)

# Stime puntuali

(mean(post_pred))
(mean(posterior_beta0))
(mean(posterior_beta1))
(mean(posterior_sigma))
(mean(posterior_xi))

# IC

post_pred_mcmc <- as.mcmc(post_pred)
print("intervallo di confidenza al 95% per la posterior predictive")
HPDinterval(post_pred_mcmc, prob = 0.95)

beta0_mcmc <- as.mcmc(posterior_beta0)
print("intervallo di confidenza al 95% per beta0")
HPDinterval(beta0_mcmc, prob = 0.95)

beta1_mcmc <- as.mcmc(posterior_beta1)
print("intervallo di confidenza al 95% per beta1")
HPDinterval(beta1_mcmc, prob = 0.95)
```

```r
sigma_mcmc <- as.mcmc(posterior_sigma)
print("intervallo di confidenza al 95% per sigma")
HPDinterval(sigma_mcmc, prob = 0.95)

xi_mcmc <- as.mcmc(posterior_xi)
print("intervallo di confidenza al 95% per xi")
HPDinterval(xi_mcmc, prob = 0.95)


return_periods <- c(10, 20, 50, 100)  # Livelli di ritorno in anni
time  <-  c(10, 20, 50, 100)
nsim <- length(post_pred)
rl_sim <- matrix(NA,4,nsim)

for (f in 1:4) {
  t <- time[f]
  post_pred_rl <- matrix(NA,nsim,t)
  for (j in 1:t) {
    for (i in 1:nsim) {
      post_pred_rl[i,j] <- rgev(1, loc = posterior_beta0[i] + posterior_beta1[i] * j,
                                scale = posterior_sigma[i],
                                shape = posterior_xi[i])
    }
  }
  post_pred_rl <- t(apply(post_pred_rl, 1, sort))
  rl_sim[f,] <- post_pred_rl[,t-1]
}

return_levels <- rowMeans(rl_sim)

# Calcola gli intervalli di confidenza empirici
credibility_intervals <- apply(rl_sim, 1, function(x) {
  lower <- quantile(x, 0.025)  # Lower 2.5% quantile
  upper <- quantile(x, 0.975)  # Upper 97.5% quantile
  c(lower, upper)
})

# Stampa risultati
return_levels
credibility_intervals <- t(credibility_intervals)  # Transpose for better readability
colnames(credibility_intervals) <- c("Lower_95%", "Upper_95%")
credibility_intervals
```

Riferimenti bibliografici

[1] Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer Series in Statistics. London: Springer-Verlag, 2001. DOI: 10.1007/978-1-4471-3675-0.