Università degli Studi di Firenze

Scuola di Ingegneria - Dipartimento di Ingegneria dell'Informazione

Relazione di Quantitative Evaluation of Stochastic Models

Smoothing di CDF attraverso BPH

Autrice
Elizaveta Mochalova

Anno Accademico 2024/2025

Indice

Introduzione **ii**

1 Implementazione **1**

1.1 Tipologie di CDF disponibili 1

1.1.1 Normal Distribution 1

1.1.2 Exponential Distribution 1

1.1.3 Uniform Distribution 2

1.1.4 Beta Distribution 2

1.1.5 Erlang Distribution 2

1.2 Bernstein Phase-Type 3

1.3 Campionamento 4

1.4 Approssimazione con BPH 5

2 Analisi dei risultati **7**

2.1 Metriche di comparazione 7

2.2 Prove eseguite 8

2.2.1 CDF diverse 8

2.2.2 Ordine del BPH 9

2.2.3 Numero di fasi dell'Erlang 10

3 Conclusioni **11**

Introduzione

Questo elaborato ha l'obiettivo di analizzare e approssimare distribuzioni cumulative di probabilità (CDF) tramite il metodo di Bernstein Phase-Type (BPH).

La CDF (Cumulative Distribution Function) è una funzione in probabilità e statistica, che associa a ogni valore reale la probabilità che una variabile casuale assuma un valore minore o uguale a quello dato. Per una variabile casuale X, la CDF è definita come:

$$F(x) = \mathbb{P}(X \leq x)$$

La CDF è una funzione crescente, limitata tra 0 e 1, che descrive completamente il comportamento probabilistico di una variabile casuale. È particolarmente utile perché fornisce informazioni cumulative sulle probabilità ed è spesso più facile da stimare e confrontare rispetto alla funzione di densità.

In questo progetto si adotta un approccio ispirato al lavoro di Babu et al. [2], che propone l'uso dei polinomi di Bernstein per ottenere stime regolari e stabili di funzioni di distribuzione e densità. I polinomi di Bernstein offrono infatti una base flessibile per costruire approssimazioni che rispettano le proprietà fondamentali delle CDF (monotonia e valori compresi tra 0 e 1).

Passi del lavoro:

- Generazione di una distribuzione cumulativa standard (esponenziale, Erlang, normale, uniforme, beta), scelta dall'utente.
- Campionamento di un numero specificato di punti casuali dalla distribuzione selezionata.
- Calcolo di un'approssimazione della CDF originale utilizzando il metodo Bernstein Phase-Type, che sfrutta polinomi di Bernstein per costruire un'approssimazione della funzione distribuzione cumulativa.
- Visualizzazione grafica sia della CDF originale che dell'approssimazione ottenuta, per un confronto diretto.
- Valutazione numerica della bontà dell'approssimazione tramite due metriche standard: Wasserstein Distance e Kolmogorov-Smirnov Distance.

Esperimenti

Per valutare la qualità dell'approssimazione e l'efficacia del metodo proposto, abbiamo condotto diversi esperimenti variando i seguenti parametri:

- **CDF diverse**: osservare se ci sono alcune CDF che sfruttano meglio l'approssimazione tramite BPH.
- **Ordine del BPH**: osservare come la scelta dell'ordine del polinomio di Bernstein cambi la qualità dell'approssimazione.
- **Numero di fasi dell'Erlang**: nel caso della distribuzione Erlang, valutare l'impatto del numero di fasi sulla forma della CDF e sull'approssimazione.

L'obiettivo di questi esperimenti è comprendere come ciascun parametro contribuisca alla complessità del modello e la precisione dell'approssimazione.

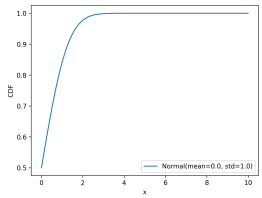
Capitolo 1

Implementazione

1.1 Tipologie di CDF disponibili

Il progetto sviluppato è in grado di creare e riprodurre più tipologie di Cumulative Distribution Functions: **Normale**, **Gaussiana**, **Esponenziale**, **Uniforme**, **Beta** o **Erlang**.

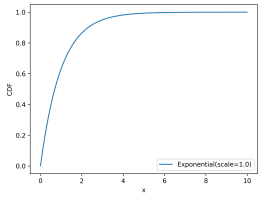
1.1.1 Normal Distribution



Questa tipologia di distribuzione è definita dai parametri media μ e deviazione standard σ, che determinano rispettivamente il centro e la dispersione della distribuzione. Entrambi i parametri possono essere definiti dall'utente ma come parametri di default si ha 0 e 1 rispettivamente. Il punto di arresto per l'asse x invece è fissato internamente ed è 10.

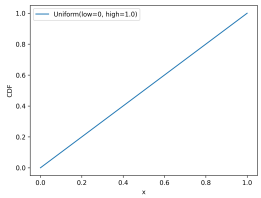
1.1.2 Exponential Distribution

Questa distribuzione è definita da λ, ovvero il tasso di arrivo (o rate). La CDF parte da 0 e tende a 1, con una curva crescente che si appiattisce. Il punto di arresto anche in questo caso per l'asse x è fissato internamente ed è 10.



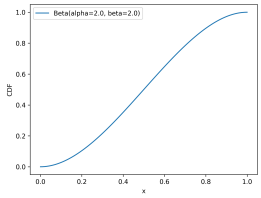
1.1.3 Uniform Distribution

Rappresenta una probabilità distribuita in modo costante tra due estremi [a,b], indicando che ogni valore ha la stessa probabilità di accadere. In questa implementazione a=0, invece b è definito dall'utente.



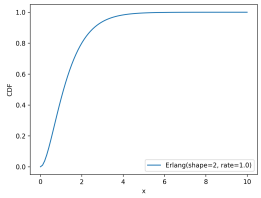
1.1.4 Beta Distribution

Rappresenta una probabilità cumulativa per una variabile casuale continua compresa tra 0 e 1, con forma controllata da due parametri positivi α e β, definiti entrambi dall'utente con i valori di default pari a 2.



1.1.5 Erlang Distribution

Questa distribuzione è una versione speciale della distribuzione gamma, usata per modellare il tempo fino al verificarsi del k-esimo evento in un processo di Poisson. Questa funzione cresce da 0 a 1 e ha una forma a S che diventa più regolare per k grandi. Il valore di k è definito dall'utente con default 2.



Per generare i dati necessari alla valutazione delle approssimazioni, è stata implementata la funzione **create_cdf**. Questa funzione consente di creare la funzione di distribuzione cumulativa (CDF) di una distribuzione scelta tra diverse disponibili e di ottenere sia i valori della CDF su un intervallo regolare. Il metodo riceve in ingresso il numero di punti da calcolare, il numero di campioni desiderati e il tipo di distribuzione con i relativi parametri. A seconda della distribuzione selezionata, vengono utilizzate le funzioni CDF delle librerie scipy.stats oppure, nel caso della distribuzione Erlang, una funzione personalizzata. A seconda della distribuzione selezionata, vengono utilizzate le funzioni CDF delle librerie scipy.stats oppure, nel caso della distribuzione Erlang, una funzione personalizzata per calcolare i valori della CDF. La funzione include inoltre la possibilità di tracciare automaticamente la CDF ottenuta, permettendo un confronto visivo immediato tra la distribuzione reale e le eventuali approssimazioni.

1.2 Bernstein Phase-Type

In generale le Bernstein Phase-Type Distributions (BPH) possono essere utilizzate per l'approssimazione di funzioni di distribuzione cumulativa (CDF). Le distribuzioni di tipo Phase-Type (PH), basate su catene di Markov continue, offrono una soluzione flessibile a questo scopo [1]: aumentando il numero di fasi, è possibile ottenere approssimazioni più accurate, accettando però un maggiore costo computazionale.

L'approssimazione dei polinomi di Bernstein per costruire CDF si basa sulla formula dei polinomi esponenziali di Bernstein:

$$\hat{F}_n(x) = \sum_{i=0}^{n} F\left(\log \frac{n}{i}\right) \binom{n}{i} e^{-ix}(1 - e^{-x})^{n-i} \tag{1.1}$$

Questo metodo ha diversi vantaggi nel nostro contesto:

- Genera delle CDF valide;
- Permette di ottenere approssimazioni flessibili per diverse distribuzioni di interesse (esponenziale, Erlang, ecc.);
- Consente un controllo diretto della qualità dell'approssimazione tramite la scelta del grado n.

1.3 Campionamento

All'interno della funzione **create_cdf** si ottiene anche un insieme di campioni utilizzati per la valutazione numerica. I campioni vengono generati in modo equidistante sull'intervallo di interesse per la maggior parte delle distribuzioni, mentre nel caso della distribuzione Erlang sono stati implementati due metodi distinti per la generazione dei campioni: il primo sfrutta direttamente la funzione `rvs` della libreria `scipy.stats`, come mostrato di seguito:

```
def sample_erlang(k, lam, size=1):
return stats.erlang.rvs(a=k, scale=1/lam, size=size)
```

Il secondo metodo genera i campioni tramite la somma di k variabili esponenziali indipendenti di parametro λ:

```
def sample_erlang_manual(k, lam, size=1):
return np.sum(np.random.exponential(scale=1 / lam, size=(size, k)), axis=1)
```

1.4 Approssimazione con BPH

La funzione `bph` implementa l'approssimazione Bernstein Phase-Type (BPH) della CDF calcolata precedentemente, sfruttando una rappresentazione basata su polinomi di Bernstein applicati a variabili esponenziali. L'input principale della funzione è un oggetto funzione `cdf_function` ottenuto tramite interpolazione lineare dei valori campionati della CDF originale, estesi includendo i valori noti agli estremi: zero all'inizio e uno alla fine del dominio. A tal fine, i campioni vengono ampliati con:

```
x_samples_extended = np.concatenate(([start], x_samples, [stop]))
cdf_samples_extended = np.concatenate(([0.0], cdf_samples, [1.0]))
sort_indices = np.argsort(x_samples_extended)
x_samples_extended = x_samples_extended[sort_indices]
cdf_samples_extended = cdf_samples_extended[sort_indices]

cdf_function = interp1d(
    x_samples_extended, cdf_samples_extended,
    kind='linear', bounds_error=False, fill_value=(0.0, 1.0)
)
```

La funzione `bph` normalizza i valori di valutazione `x_values` nel dominio $[0, 1]$, calcola il corrispondente dominio logaritmico negativo e valuta la somma pesata delle basi di Bernstein esponenziali:

```
n = order
x_normalized = (x_values - start) / (stop - start)
x_normalized = np.clip(x_normalized, 1e-10, 1 - 1e-10)
t_values = -np.log(x_normalized)

bernstein_bph = np.zeros_like(x_values, dtype=float)
bernstein_bph[x_values = start + 1e-10] = 0.0
bernstein_bph[x_values = stop - 1e-10] = 1.0
```

```
for i in range(1, n + 1):
    eval_point = start + (stop - start) * np.exp(-np.log(i / n))
    eval_point = np.clip(eval_point, start, stop)
    cdf_val = cdf_function(eval_point)

    binom_coeff = comb(n, i, exact=False)
    bernstein_basis = binom_coeff * np.exp(-i * t_interior)
        * (1 - np.exp(-t_interior)) ** (n - i)

    bernstein_bph[interior_mask] += cdf_val * bernstein_basis
```

Questa procedura produce una CDF approssimata continua e crescente, coerente con la funzione originale, e consente di controllare la qualità dell'approssimazione tramite l'ordine `n` scelto.

Capitolo 2

Analisi dei risultati

2.1 Metriche di comparazione

Per valutare la qualità dell'approssimazione della CDF ottenuta tramite Bernstein Phase-Type, sono state implementate due metriche di confronto: la Wasserstein Distance e la Kolmogorov–Smirnov Distance.

- **wasserstein_distance**

  La funzione calcola una stima numerica della distanza di Wasserstein tra due distribuzioni, che rappresenta la "quantità di lavoro" necessaria per trasformare una distribuzione nell'altra. L'implementazione si basa sulla valutazione del valore assoluto della differenza tra le due CDF.

- **kolmogorov_smirnov_distance**

  Questa invece calcola la distanza di Kolmogorov–Smirnov, definita come la massima differenza assoluta tra due funzioni di distribuzione cumulativa su un dato intervallo. L'implementazione effettua un campionamento uniforme di punti sull'intervallo di interesse, valuta entrambe le CDF sui punti selezionati e restituisce la massima differenza rileva-

ta. Questa metrica fornisce un'indicazione diretta del punto di massimo scostamento tra le distribuzioni ed è particolarmente utile per verificare la bontà dell'approssimazione in modo semplice e immediato.

2.2 Prove eseguite

Come anticipato nell'introduzione, vogliamo confrontare i risultati ottenuti variando alcuni parametri attraverso una serie di esperimenti.

2.2.1 CDF diverse

Eseguendo le prove con un numero fisso di campioni n=30 e ordine di BPH m=8 si ottengono i seguenti risultati:

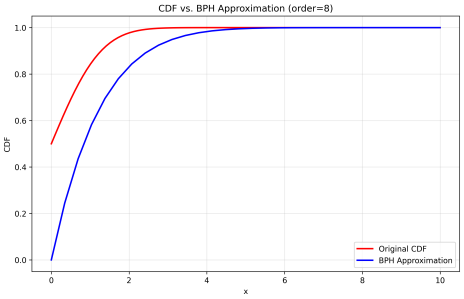


Figura 2.1: Normal(0, 1)

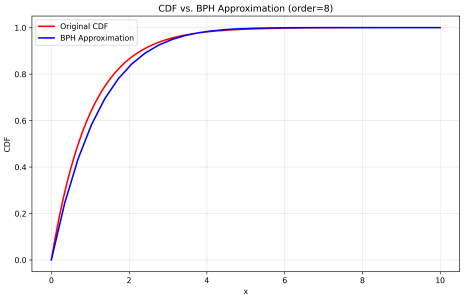


Figura 2.2: Exp(1)

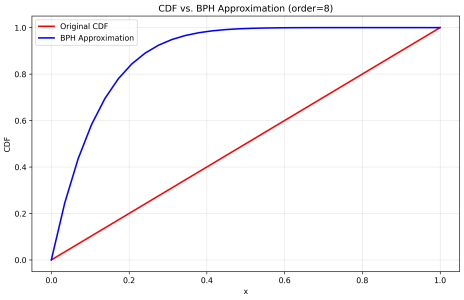


Figura 2.3: Uniform(a=0, b=1)

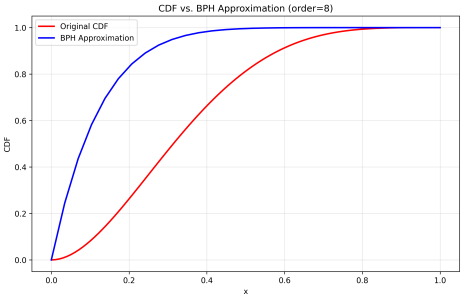


Figura 2.4: Beta(α=2, β=4)

I risultati possono essere riassunti in base alle metriche definite precedentemente.

Distribuzione	Parametri (n, m)	Wasserstein	Kolmogorov-Smirnov
Normal	30, 8	0.72	0.50
	125, 25	0.18	0.50
Exponential	30, 8	0.12	0.06
	125, 25	0.61	0.34
Uniform	30, 8	0.39	0.65
	125, 25	0.46	0.84
Beta	30, 8	0.22	0.57
	125, 25	0.29	0.85
Erlang	30, 8	0.92	0.31
	125, 25	1.62	0.68

Tabella 2.1: Distanze Wasserstein e Kolmogorov-Smirnov per varie distribuzioni e parametri

2.2.2 Ordine del BPH

Fissando il numero dei campioni al variare dell'ordine di BPH si ottengono: Inoltre per quanto riguarda Wasserstein Distance e Kolmogorov-Smirnov Distance:

- n=30, m=9 - WD 1.07 , KSD 0.35
- n=30, m=30 - WD 1.76 , KSD 0.73

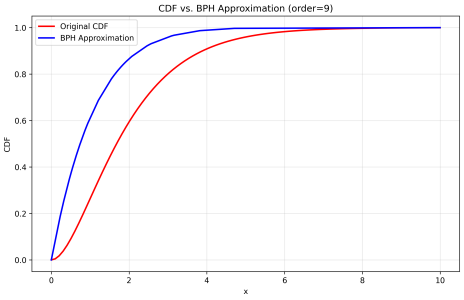


Figura 2.5: Erlang(k=1, λ=1.0) con n=30, m=9

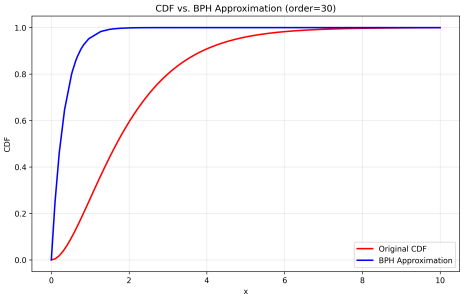


Figura 2.6: Erlang(k=5, λ=1.0) con n=30, m=30

2.2.3 Numero di fasi dell'Erlang

Si eseguono delle prove con la distribuzione di Erlang variando il valore di k con valore fisso λ=1.0.

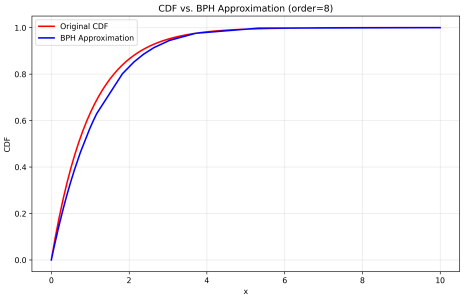


Figura 2.7: Erlang(k=1, λ=1.0) con n=30, m=8

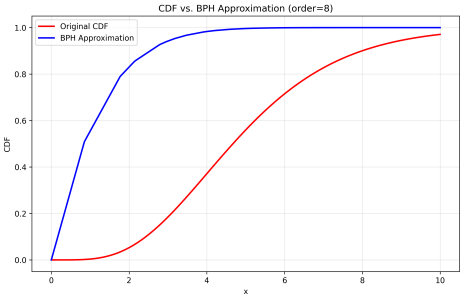


Figura 2.8: Erlang(k=5, λ=1.0) con n=30, m=8

Le metriche ottenute in questo caso sono:

- k=1 - WD 0.13 , KSD 0.06
- k=5 - WD 3.79 , KSD 0.79

Capitolo 3

Conclusioni

Il progetto ha dimostrato l'efficacia dell'approssimazione delle funzioni di distribuzione cumulativa tramite il metodo Bernstein Phase-Type (BPH). Gli esperimenti su diverse tipologie di CDF — tra cui Normale, Esponenziale, Uniforme, Beta ed Erlang — hanno confermato che il metodo riesce a fornire approssimazioni stabili e regolari, preservando le proprietà fondamentali di una CDF, come la monotonia e il range [0, 1].

I risultati hanno mostrato che:

- La CDF esponenziale è quella per cui l'approssimazione con BPH fornisce i risultati migliori, probabilmente per la sua struttura semplice e monotona.
- A parità di numero di campioni, utilizzare un ordine del polinomio più basso tende a produrre approssimazioni migliori e più stabili, suggerendo che un aumento eccessivo dell'ordine possa introdurre oscillazioni o peggiorare l'aderenza ai dati.
- Per la distribuzione Erlang, l'approssimazione risulta più efficace quando il numero di fasi k è piccolo; al crescere di k, infatti, la complessi-

tà della forma della CDF rende l'approssimazione più difficile e meno precisa.

Le metriche di valutazione — distanza di Wasserstein e test di Kolmogorov-Smirnov — hanno confermato l'impatto di questi fattori sulla qualità delle approssimazioni, permettendo di osservare in modo quantitativo come le scelte di modello influenzino le prestazioni.

Il metodo BPH si è quindi rivelato flessibile e controllabile, ma la sua efficacia dipende strettamente dalle caratteristiche della distribuzione da approssimare e dai parametri scelti per l'approssimazione.

Bibliografia

[1] Andras Horvath, Illes Horvath, Marco Paolieri, Miklos Telek, and Enrico Vicario. Approximation of cumulative distribution functions by bernstein phase-type distributions.

[2] G. Jogesh Babu, Angelo J. Canty, and Yogendra P. Chaubey. Application of bernstein polynomials for smooth estimation of a distribution and density function. *Journal of Statistical Planning and Inference*, 105(2):377–392, 2002.