



# Reti Bayesiane a Tempo Continuo: generazione di reti e campionamento di traiettorie

---

Pietro Epis  
Matricola 845045

A.A. 2020/2021

Relatore: Prof. Fabio Stella  
Correlatore: Dott. Alessandro Bregoli

# CTBN - Introduzione

Modello grafico-probabilistico

Rappresentare sistemi dinamici il cui stato progredisce nel tempo (continuo)

Superare i limiti delle Dynamic Bayesian Networks

Basate su Continuous Time Markov Processes e Bayesian Networks

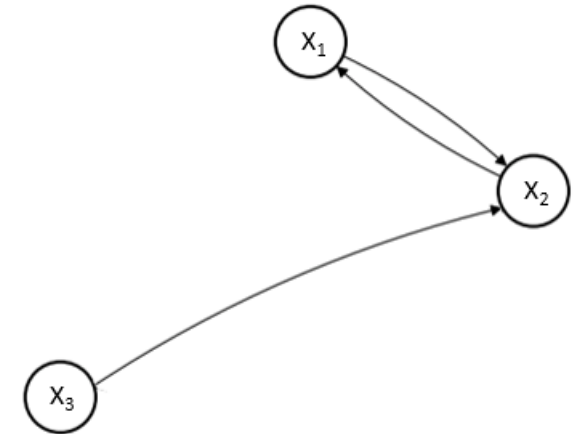
# CTBN - Definizione

Sia  $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$  un **insieme di variabili**, dove ciascuna variabile rappresenta un CTMP

Una CTBN  $\mathcal{N}$  su  $X$  è definita da:

- Una **distribuzione iniziale**  $\mathcal{B}_0$  definita come rete Bayesiana
- Un **modello di transizione** costituito da:
  - Un grafo orientato  $\mathcal{G}$  sulle variabili in  $X$
  - Una Conditional Intensity Matrix  $Q_{X | Pa(X)}$  per ogni variabile in  $X$

È ammissibile che il grafo  $\mathcal{G}$  contenga cicli



# Conditional Intensity Matrix (CIM)

L'elemento  $q_{ij}$  rappresenta la probabilità istantanea che  $X$  effettui una transizione dallo stato  $i$  allo stato  $j$

Le matrici variano al variare del tempo del tempo  $t$ , in funzione dei valori che le variabili in  $\mathbf{P} = \mathbf{Pa}(X)$  assumono all'istante  $t$

Le CIM esprimono quindi le dipendenze tra le variabili

$$Q_{X|P} = \begin{bmatrix} -q_1(P) & q_{12}(P) & \dots & q_{1n}(P) \\ q_{21}(P) & -q_2(P) & \dots & q_{2n}(P) \\ \vdots & \vdots & \ddots & \vdots \\ q_{n1}(P) & q_{n2}(P) & \dots & -q_n(P) \end{bmatrix}$$

# Traiettoria

Una specifica istanza dei valori di una variabile  $X$ , per ogni istante temporale  $t$ , prende il nome di **traiettoria**

Una ed una sola variabile può effettuare una transizione verso un nuovo stato in uno specifico istante

	Time	$X_1$	$X_2$	$X_3$
0	0	0	0	0
1	0.0494	2	0	0
2	1.5477	2	1	0
3	1.692	2	1	2
4	1.7851	1	1	2
5	1.9497	2	1	2
6	2.2106	2	2	2
7	2.3166	2	2	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
993	597.96	2	1	0
994	598.155	0	1	0
995	598.696	0	0	0
996	598.856	0	0	2
997	599.426	0	2	2
998	599.58	1	2	2
999	601.761	2	2	2
1000	602.012	-1	-1	-1

# PyCTBN

Libreria **Open-Source** per lavorare con il modello delle CTBN

Realizzata dal **MAD Lab** (Models and Algorithms for Data and Text Mining)

Sviluppata in **Python**, sfruttando librerie quali Pandas, NumPy e NetworkX

Progettata per essere estendibile ed efficiente

# Campionamento di traiettorie - TrajectoryGenerator

La classe `TrajectoryGenerator` implementa il campionamento nel metodo:

`CTBN_Sample(t_end, max_tr)`

Condizioni di terminazione:

- Tempo massimo
- Numero di transizioni prestabilito

La traiettoria viene ritornata come oggetto `pandas.DataFrame`

Determina  $\text{Time}(X) \forall X \in \mathbf{X}$

Seleziona  $X_{tr} = \arg \min_{Y \in \mathbf{X}} \text{Time}(Y)$

Determina il nuovo valore di  $X_{tr}$

Aggiungi la transizione alla traiettoria

Imposta  $\text{Time}(X) = \text{undefined} \forall X \in X_{tr} \cup \text{Cd}(X_{tr})$

# TrajectoryGenerator - Multiprocessing

Realizzato tramite il package `multiprocessing` di Python

Ogni processo si occupa di generare una singola traiettoria

Le traiettorie vengono raggruppate in una `lista condivisa`

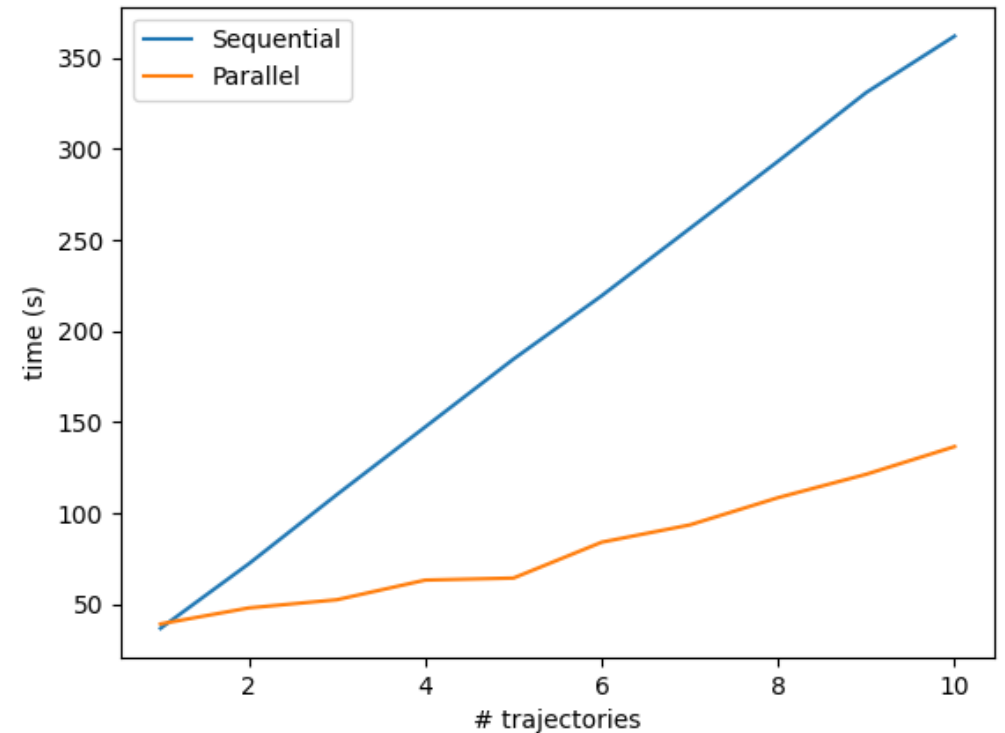
## Esempio efficientamento:

- 20 variabili ternarie, densità 0.4
- 10 traiettorie di 1000 osservazioni

Sequenziale: 361.8562 s

Parallelo 4 core: 136.6533 s

↓ - 62.24 %





# Generazione di reti - NetworkGenerator

Generazione casuale del **grafo** e delle **CIM** specificando il numero di variabili e la relativa cardinalità

**generate\_graph**(density, fixed)

Stabilire per ciascuna coppia di variabili (X, Y) se esiste un arco da X a Y e viceversa

Estrazione valore casuale (0 o 1) da una **Bernoulliana** di parametro density

**generate\_cims**(min\_val, max\_val)

Gli argomenti stabiliscono il range entro cui estrarre i coefficienti

**Generazione della riga di una matrice:**

- Estrazione da distribuzione **uniforme** di tutti i valori
- Normalizzazione in modo che  $\sum_{j \neq i} q_{ij} = |q_i|$

# Esportazione dei dati - AbstractExporter

Esportazione e salvataggio su file di strutture e traiettorie

Utilizzo dei file per l'interazione tra i moduli della libreria

La classe astratta **AbstractExporter** raggruppa i dati che descrivono la rete, a prescindere dal formato

Il metodo astratto **out\_file** viene implementato in modo specifico nelle varie realizzazioni, come **JSONExporter** nel caso del formato JSON

```
{
  "variables ": [
    {" Name ": "X", " Value ": 3},
    {" Name ": "Y", " Value ": 3},
    {" Name ": "Z", " Value ": 3}
  ],
  "dyn.str ": [
    {" From ": "X", "To ": "Z"},
    {" From ": "Y", "To ": "X"},
    {" From ": "Z", "To ": "X"}
  ],
  "dyn. cims ": {
    "X": {
      "Y=0,Z =0": [
        {"0": -2.5226 , "1": 2.495 , "2": 0.0276} ,
        {"0": 1.822 , "1": -2.8677 , "2": 1.0458} ,
        {"0": 0.1172 , "1": 2.4934 , "2": -2.6106}
      ], ...
    },
    "Y": { ... },
    "Z": { ... },
  },
  "samples": [ ... ]
}
```

# Conclusioni e Sviluppi Futuri

Proprietà di **Faithfulness**

Implementazione con linguaggi più performanti

The background of the slide is a complex, abstract network diagram. It consists of numerous nodes of varying sizes and colors (dark blue, light blue, and grey) connected by thin, light grey lines. Some nodes are highlighted with larger, concentric circles. The overall aesthetic is modern and technological.

**Grazie per l'attenzione!**

# Bibliografia

U. Nodelman. Continuous Time Bayesian Networks. Master's thesis, Stanford University, 2007

R. Diestel. Graph Theory. Springer, 2000

A. P. Dawid. Conditional independence in statistical theory. Journal of the Royal Statistical Society. Series B (Methodological), 41:1 - 31, 1979

I. Ben-Gal. Bayesian networks. In F. Ruggeri, F. Faltin, and R. Kenett, editors, Encyclopedia of Statistics in Quality and Reliability. Wiley & Sons, 2007

S. M. Ross. Stochastic Processes. Wiley, 2nd edition, 1996

C. Shelton and G. Ciardo. Tutorial on Structured Continuous-Time Markov Processes. Journal of Artificial Intelligence Research, page 725 - 778, 12 2014

O. Haggstrom. Finite Markov Chains and Algorithmic Applications. London Mathematical Society Student Texts, 2002

U. Nodelman, C. Shelton, and D. Koller. Continuous Time Bayesian Networks. Uncertainty in artificial intelligence, pages 378 - 387, 2002

# Bibliografia

W. McKinney. Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, pages 56 - 61, 01 2010

C. R. Harris, K. J. Millman, S. J. van der Walt, et al. Array programming with numpy. Nature, pages 357 - 362, 09 2020

C. Shelton, Y. Fan, W. Lam, J. Lee, and J. Xu. Continuous Time Bayesian Network Reasoning and Learning Engine. Journal of Machine Learning Research, 03 2010

H. Boudali and J.B. Dugan. A continuous-time bayesian network reliability modeling, and analysis framework. IEEE Transactions on Reliability, 55:86 - 97, 2006

S. M. Ross. Introduction to Probability and Statistics for Engineers and Scientists. Elsevier Science, 3rd edition, 2004

Kayvan Sadeghi. Faithfulness of probability distributions and graphs. Journal of Machine Learning Research, pages 1 - 29, 2017

M. Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley, 3rd edition, 2003