

# Learning of Bayesian networks with missing data



---

**Presentazione della tesi di laurea magistrale di:**

*Andrea Ruggieri*

Matricola: 806808

Email: [a.ruggieri4@campus.unimib.it](mailto:a.ruggieri4@campus.unimib.it)

**Relatori:**

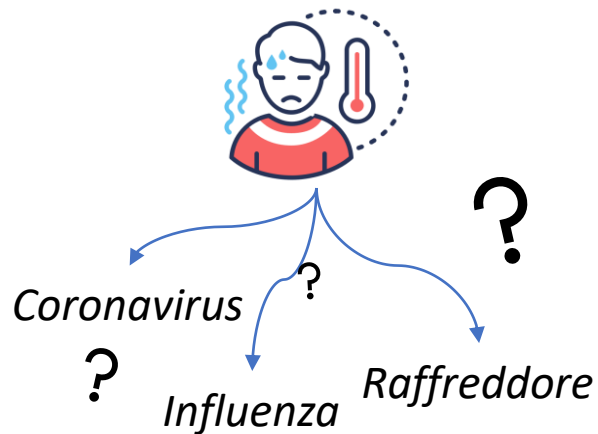
Relatore: *Prof. Fabio Stella*

Co-relatore: *Ph.D. Alessandro Bregoli*

# La presenza di valori mancanti in un dataset è problematica

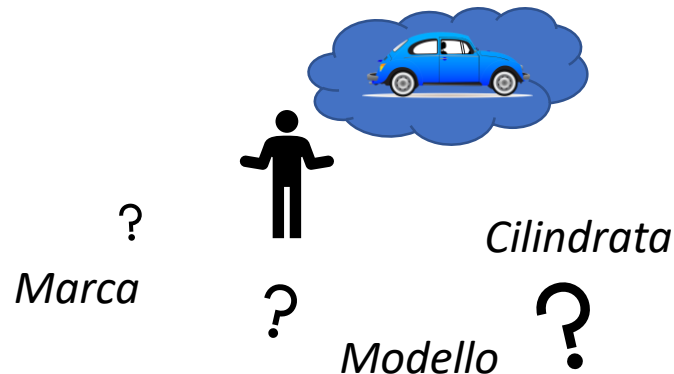
## Assenza di informazioni

Alcune informazioni risultano essere cruciali ai fini di prendere decisioni. Inoltre, l'assenza di alcuni dati potrebbe rendere impossibile la risoluzione del problema.



## Difficoltà nel predire i valori mancanti

Alcuni tipi di dati mancanti sono complicati da rimpiazzare in quanto l'unico modo è avere una corretta comprensione e conoscenza del dominio del contesto esaminato.



## Bias nei dati e nei risultati

I dati risultano essere distorti dalla realtà. Inoltre, i modelli di machine learning non funzionano o esibiscono una perdita di efficacia in presenza di dati mancanti.

*When we deal with low data quality, even the results are of low quality.*

*Garbage in, garbage out*

Nelle  
prossime  
slide...



INTRODUZIONE  
ALL'ALGORITMO EM



IMPLEMENTAZIONE



SPERIMENTAZIONE



RISULTATI E  
CONCLUSIONI



SVILUPPI FUTURI



# STATO DELL'ARTE E OBIETTIVI

## INTRODUZIONE ALL'ALGORITMO EM

**1998** Friedman pubblicò *the Bayesian Structural EM Algorithm*: l'obiettivo è apprendere la struttura di una rete Bayesiana su dati mancanti o variabili nascoste.

**2010 – 2016** Diversi studiosi dimostrarono l'efficacia dell'algoritmo EM

**2009** Friedman e Koller pubblicano il libro *Probabilistic Graphical Models: Principles and Techniques*. Si fa riferimento all'algoritmo EM per rimpiazzare i valori mancanti fissata la struttura della rete.

### Limite

Gli studiosi hanno focalizzato la loro attenzione su una versione semplificata dell'algoritmo EM che prende il nome di *EM Hard-assignment* o *EM Hard*. I risultati della versione completa (*EM Soft-Assignment* o *EM Soft*) dell'algoritmo EM sono solo teorici.

### Gli obiettivi di questo lavoro sono:

1. Progettare e sviluppare un pacchetto R che renda disponibile il codice per le diverse versioni dell'algoritmo EM.
2. Valutare e sperimentare le diverse versioni dell'algoritmo EM in diverse condizioni sperimentali e per diverse misure di prestazione.
3. Fornire al lettore indicazioni utili qualora esso abbia l'esigenza di rimpiazzare valori mancanti.



# L'ALGORITMO EM

## INTRODUZIONE ALL'ALGORITMO EM

Begin procedure *EM\_Algorithm*(...){

$\forall \theta \in BN$  fissa un valore iniziale  $\hat{\theta}_0$

**while** ( $|\hat{\theta}_{t-1} - \hat{\theta}_t| < \epsilon$ ){

**Expectation step:** l'algoritmo usa i parametri correnti  $\theta^t$  per computare le expected sufficient statistics:

- Per ogni dato  $o$ , vengono compute tutte le probabilità marginali:

$$Q(X, U) = P(X, U | o, \theta)$$

- Calcolo delle expected sufficient statistics:

$$\bar{M}_{\theta}[u] = \sum_{m=1}^M \sum_{h[m] \in \text{Val}(H[m])} Q(h[m]) I\{\xi[m] < Y = y\}$$

**Maximisation step:** vengono utilizzate le ESS per eseguire la maximum likelihood estimation:

$$\theta_{x|u}^{t+1} = \frac{\bar{M}_{\theta^t}[x, u]}{\bar{M}_{\theta^t}[u]}$$

}

}

### Considerazioni:

- L'inizializzazione dei parametri risulta essere cruciale e può avvenire in diversi modi.
- Il passo di Maximisation risulta essere lineare.
- L'algoritmo EM garantisce sempre la convergenza



# EM HARD E EM SOFT A CONFRONTO

## INTRODUZIONE ALL'ALGORITMO EM

---

### Algorithm 1: The Soft version of the EM algorithm.

---

Choose an initial value  $\hat{\theta}_0$  for  $\theta$ ;

**while**  $|\hat{\theta}_{j-1} - \hat{\theta}_j| < \varepsilon$ , *increasing j*: **do**

$\hat{\theta}_j = \hat{\theta}_{j-1}$ ;

**Expectation step**: all possible cases of completion of the missing data are computed:  $Q(X, U) = P(X, U|o, \theta)$ ;

**ESS**: We use the marginal probabilities to compute:

$$\bar{M}_\theta[u] = \sum_{m=1}^M \sum_{h[m] \in \text{Val}(H[m])} Q(h[m]) I\{\xi[m] < Y \geq y\};$$

**Maximisation step**: compute the new estimate  $\hat{\theta}_j$  given the ESS;

**end**

Estimate  $\theta$  with the last  $\hat{\theta}_j$ .

---

---

### Algorithm 2: The Hard version of the EM algorithm.

---

Choose an initial value  $\hat{\theta}_0$  for  $\theta$ .

**while**  $|\hat{\theta}_{j-1} - \hat{\theta}_j| < \varepsilon$ , *increasing j*: **do**

$\hat{\theta}_j = \hat{\theta}_{j-1}$ .

**Expectation step**: all possible cases of completion are computed but it selects the single assignment which maximizes:  $P(h|o[m], \theta^t)$ .

**ESS**: We use the selected marginal probability to compute:

$$\bar{M}_\theta[u] = \sum_{m=1}^M I\{\xi[m] < X \geq x^1\}.$$

**Maximisation step**: compute the new estimate  $\hat{\theta}_j$  given the ESS;

**end**

Estimate  $\theta$  with the last  $\hat{\theta}_j$ .

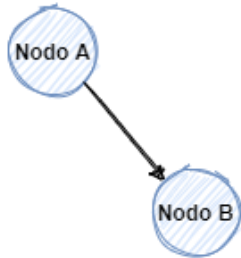
---



# APPROCCIO METODOLOGICO

## IMPLEMENTAZIONE

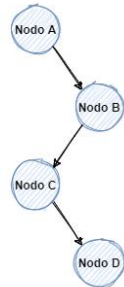
### Step 1



L'algoritmo EM funziona su un esempio elementare costituito da soli due nodi binari.

Validazione dei risultati: calcoli manuali.

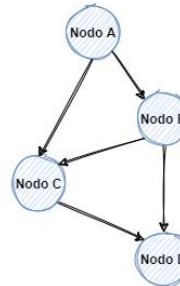
### Step 2



L'algoritmo EM funziona su reti semplici. Ogni nodo ha al massimo un genitore e/o un figlio. Tutti i nodi sono binari.

Validazione dei risultati: calcoli manuali e test di unità.

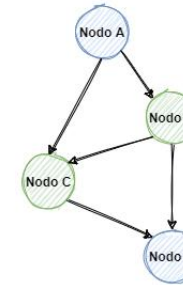
### Step 3



L'algoritmo EM funziona su reti complesse. Non ci sono limiti alla struttura della rete. Tutti i nodi sono binari.

Validazione dei risultati: test di unità.

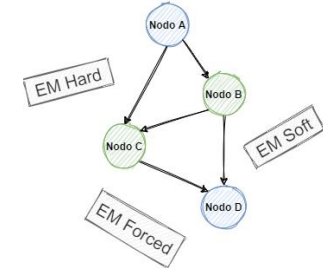
### Step 4



L'algoritmo EM funziona su reti complesse. Non ci sono limiti alla struttura della rete. I nodi possono assumere valori discreti.

Validazione dei risultati: test di unità.

### Step 5



Viene fornita l'implementazione di EM Hard e di EM Forced. L'algoritmo funziona su tutti i tipi di rete.

Validazione dei risultati: test di unità e calcoli manuali.



# FORMULAZIONE DI UN ESPERIMENTO

SPERIMENTAZIONE

Selezione e analisi delle proprietà del dataset



Definizione del numero di **repliche**

Scelta delle **dimensionalità** dei subset (o sample size)

Individuazione delle **variabili** che possono contenere valori mancanti

Identificazione del **tipo** (MCAR, MAR o MNAR) di dato mancante

Scelta della **frequenza** di valori mancanti da generare



Generazione dei dataset parziali utilizzando il metodo `ampute` di Mice.





# TEST ESEGUITI

## SPERIMENTAZIONE

- Sulla base della **grandezza della rete**, l'algoritmo EM è stato eseguito su reti di *piccole* (2-19 nodi), *medie* (20-49 nodi) e *grandi* (>50 nodi) dimensioni.
- Sulla base del **tipo e distribuzione di dato mancante**, l'algoritmo EM è stato sperimentato su dati *MCAR*, *MAR* e *MNAR*. I valori mancanti sono stati generati *uniformemente su tutto il dataset* o focalizzando l'attenzione su *specifici nodi* (nodi foglia, nodi radice, nodi più connessi...).
- Sulla base della **frequenza**, l'algoritmo EM è stato sperimentato su dataset che esibiscono una *rara* ( $\leq 1\%$ ), *bassa* ( $\leq 3\%$ ), *media* ( $\leq 10\%$ ) e *alta* ( $> 10\%$ ) frequenza di valori mancanti.

Dataset	Network size	Distribution	Frequency of missing values	replicates	Number of data
Asia	Small	Random patterns MNAR e MCAR	low	10	100; 200; 300; 400; 500; 1000; 1500; 2000
			medium	10	100; 200; 300; 400; 500; 1000; 1500; 2000
			high	10	100; 200; 300; 400; 500; 1000; 1500; 2000
Sports	Small	Random patterns MNAR e MCAR	low	10	100, 200, 400, 800, 1200, 1600, 5000
			high	10	100, 200, 400, 800, 1200, 1600
		Most central nodes	low	10	100, 200, 400, 800, 1200, 1600, 2000
			high	10	100, 200, 400, 800, 1200, 1600
Alarm	Medium	Random patterns MNAR e MCAR	low	8	200; 400; 600; 1000; 1500
			medium	8	200; 400; 600; 1000; 1500
		Most central nodes	low	8	200; 400; 600; 1000; 1500
			medium	8	200; 400; 600; 1000; 1500
Property	Medium	Random patterns MNAR e MCAR	low	8	200, 400, 800, 1100
			medium	8	400, 800, 1100
		Most central nodes	low	8	200, 400, 800, 1100
		Leaves	low	8	200, 400, 800, 1100
ForMed	Large	Random patterns MNAR	rare	8	300, 600, 1000, 1400
		Roots	low	8	300, 600, 1000, 1400
		With major outdegree	rare	8	300, 600, 1000, 1400
		Leaves	low	8	300, 600, 1000, 1400
		Random patterns MCAR	low	8	300, 600, 1000, 1400
		Most central nodes	low	8	300, 600, 1000, 1400
Pathfinder	Large	Random patterns MNAR	rare	8	300, 600, 1000, 1400
			low	8	1000
		Most central nodes	low	8	300,600,1000, 1400
		With major indegree	rare	8	300,600,1000
		With major outdegree	rare	8	300,600,1000
		leaves	rare	8	300,600,1000
		Random patterns MCAR	rare	8	300,600,1000
Hailfinder	Large	Random patterns MNAR	low	8	300, 600, 900, 1200
			rare	8	300, 600, 900, 1200
		Random patterns MCAR	rare	8	300, 600, 900, 1200
		Most central nodes	low	8	300, 600, 900, 1200
		Leaves	low	8	300, 600, 900, 1200

Esperimenti eseguiti



# APPROCCIO METODOLOGICO

## SPERIMENTAZIONE

Una **replica** termina quando *EM Hard*, *EM Soft* e *EM Forced* hanno terminato la computazione per il dataset parziale  $D_i$  associato a  $rep_i$ .

Un' **iterazione** termina quando per un specifico sample size, tutte le repliche vengono completate.

Un **esperimento** è la computazione di tutte le iterazioni.

<i>Iteration 1</i> Esperimento dataset ASIA, random patterns, tipo MCAR							
100 data	200 data	300 data	400 data	500 data	1000 data	1500 data	2000 data
rep1	rep1	rep1	rep1	rep1	rep1	rep1	rep1
rep2	rep2	rep2	rep2	rep2	rep2	rep2	rep2
rep3	rep3	rep3	rep3	rep3	rep3	rep3	rep3
rep4	rep4	rep4	rep4	rep4	rep4	rep4	rep4
rep5	rep5	rep5	rep5	rep5	rep5	rep5	rep5
rep6	rep6	rep6	rep6	rep6	rep6	rep6	rep6
rep7	rep7	rep7	rep7	rep7	rep7	rep7	rep7
rep8	rep8	rep8	rep8	rep8	rep8	rep8	rep8
rep9	rep9	rep9	rep9	rep9	rep9	rep9	rep9
rep10	rep10	rep10	rep10	rep10	rep10	rep10	rep10

*Esempio di esperimento condotto sul dataset ASIA*

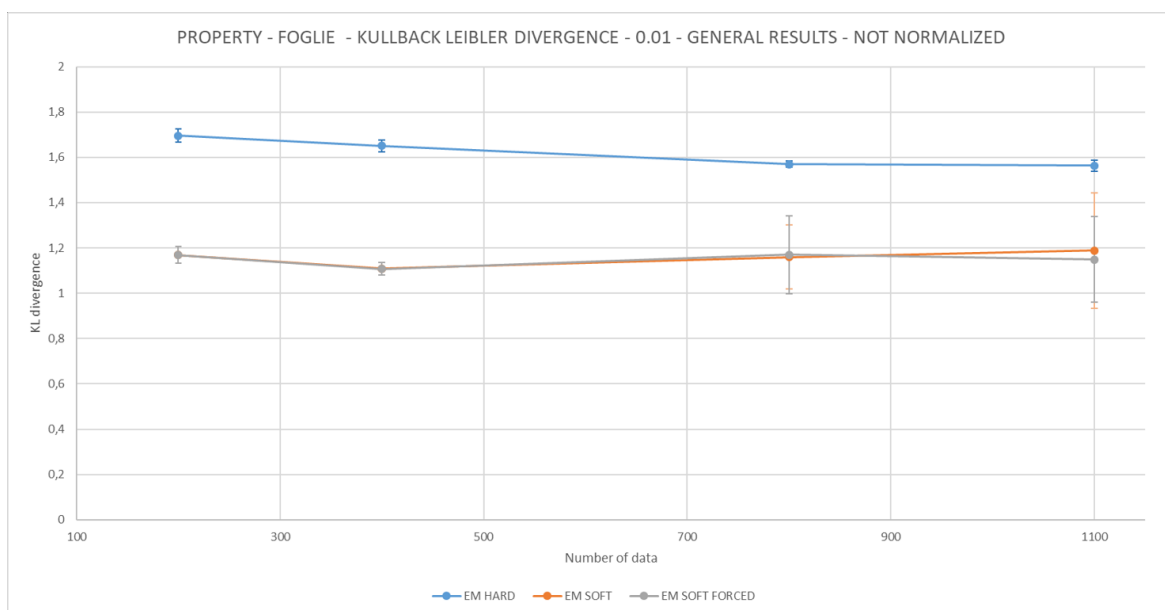


# VALUTAZIONE DEI RISULTATI

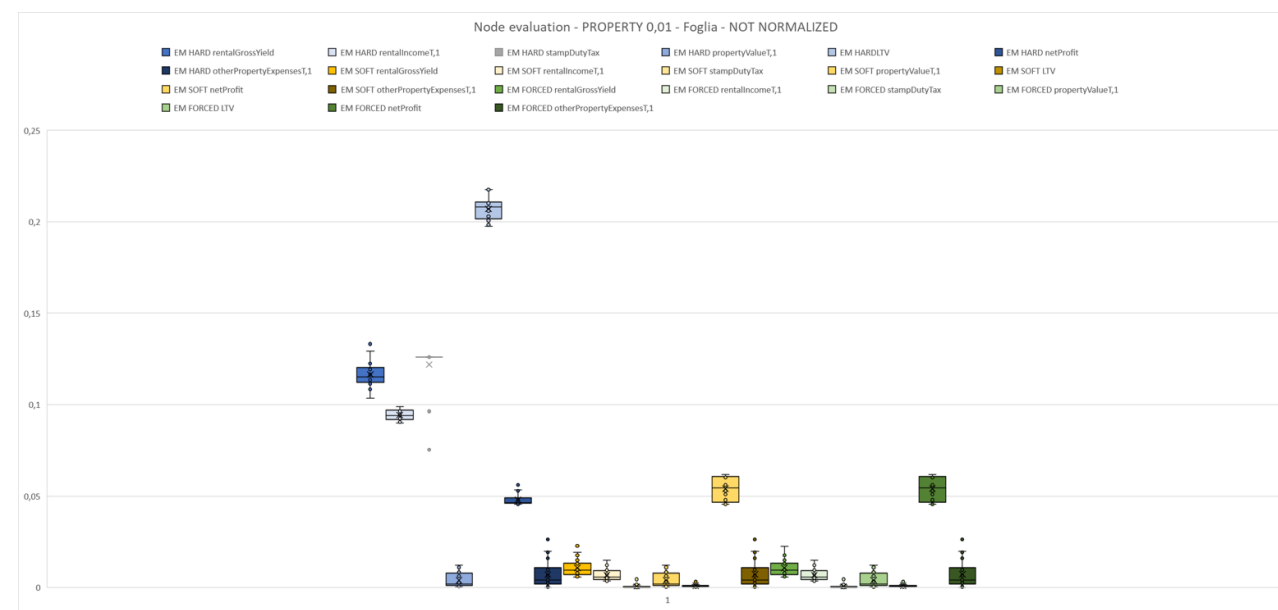
## SPERIMENTAZIONE

Alla fine di ogni esperimento, sono stati ricavati due tipi di risultati:

- **Risultati generali** derivanti dal calcolo della media e dalla deviazione standard al termine di ogni iterazione;
- **Analisi node-by-node** che ha lo scopo di indentificare eventuali nodi appresi in modo errato.



Esempio risultati generali ricavati da un esperimento condotto sulla rete Property.

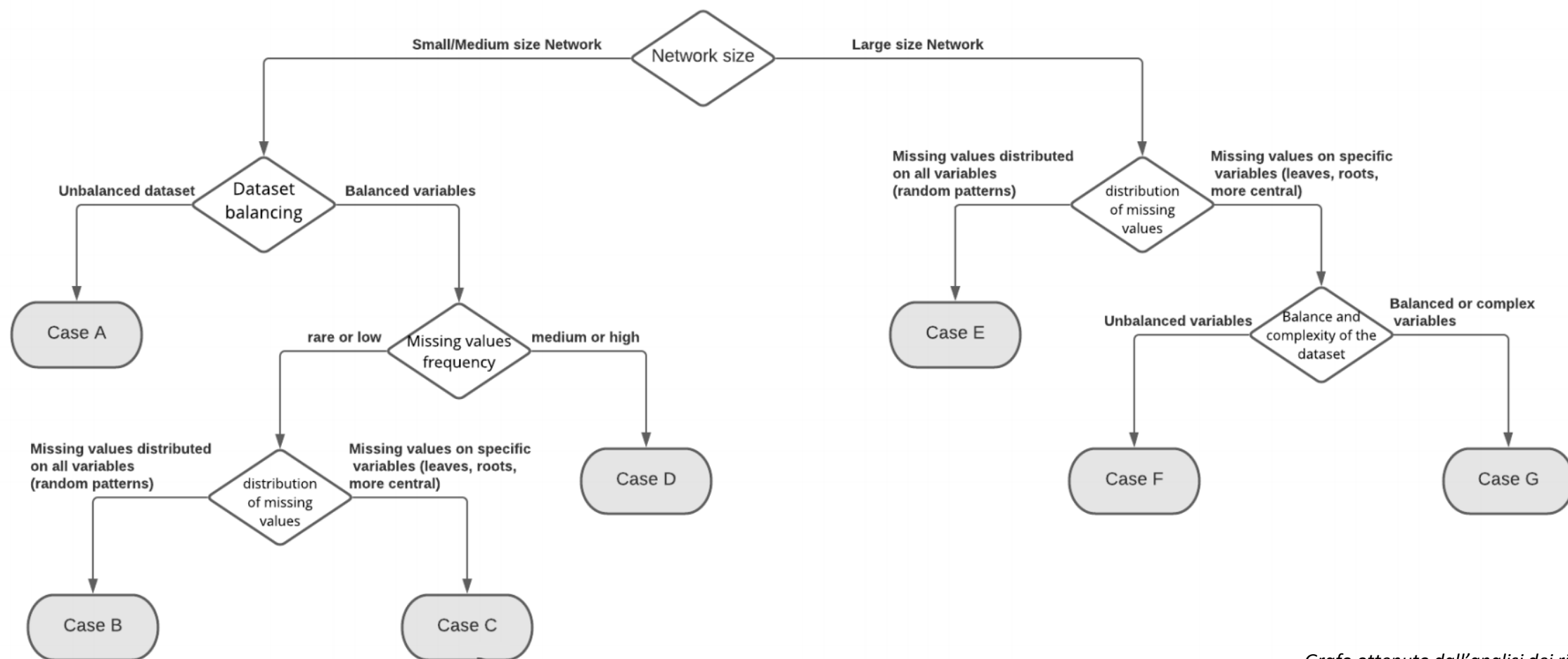


Esempio di node-by-node analysis riferito ai risultati mostrati a sinistra.



# VALUTAZIONE

## RISULTATI E CONCLUSIONI





# VALUTAZIONE

## RISULTATI E CONCLUSIONI

Caso presentato	Algoritmo consigliato	Numero iterazioni consigliate	Breve motivazione	Dataset validato
A	EM HARD, EM SOFT	3,4	L'aumento del numero di iterazioni e di valore mancanti non produce differenze statisticamente significative e i due algoritmi tendono ad imparare esattamente la stessa distribuzione di probabilità. Per motivi computazionali, è consigliabile impostare un numero di iterazioni basso.	ASIA ALARM
B	EM HARD	3	<i>EM Hard</i> tende ad avere un andamento stabile e con deviazione standard più accentuata. Al contrario, <i>EM Soft</i> tende ad esibire fenomeni di overfitting.	SPORTS PROPERTY
C	EM SOFT	3,4,5	<i>EM Hard</i> potrebbe non imparare correttamente la distribuzione uniforme dei dati e quindi apprendere meno efficacemente la distribuzione di probabilità marginale delle variabili.	SPORTS PROPERTY
D	EM HARD	3,4,5	All'aumentare dei dati, la divergenza di <i>EM SOFT</i> tende ad avvicinarsi a quella di <i>EM Hard</i> . <i>EM Hard</i> garantisce una minore deviazione standard e, conseguentemente, una minore volatilità dei risultati.	SPORTS PROPERTY
E	EM HARD	3, tante	Il fenomeno di overfitting si aggrava in <i>EM Soft</i> quanto più si aumenta il numero di iterazioni e dei valori mancanti	FORMED HAILFINDER PATHFINDER
F	EM HARD	4, tante	<i>EM Hard</i> risulta essere preferibile in termini di tempo computazionale. Anche la varianza risulta essere più contenuta, riducendo il rischio di overfitting.	FORMED HAILFINDER PATHFINDER
G	EM SOFT	5, tante	<i>EM Hard</i> fatica ad apprendere correttamente le variabili complesse (64 possibili valori). <i>EM Soft</i> risulta essere più preciso e anche la divergenza risulta essere significativamente migliore.	PATHFINDER



# ULTERIORI CONSIDERAZIONI

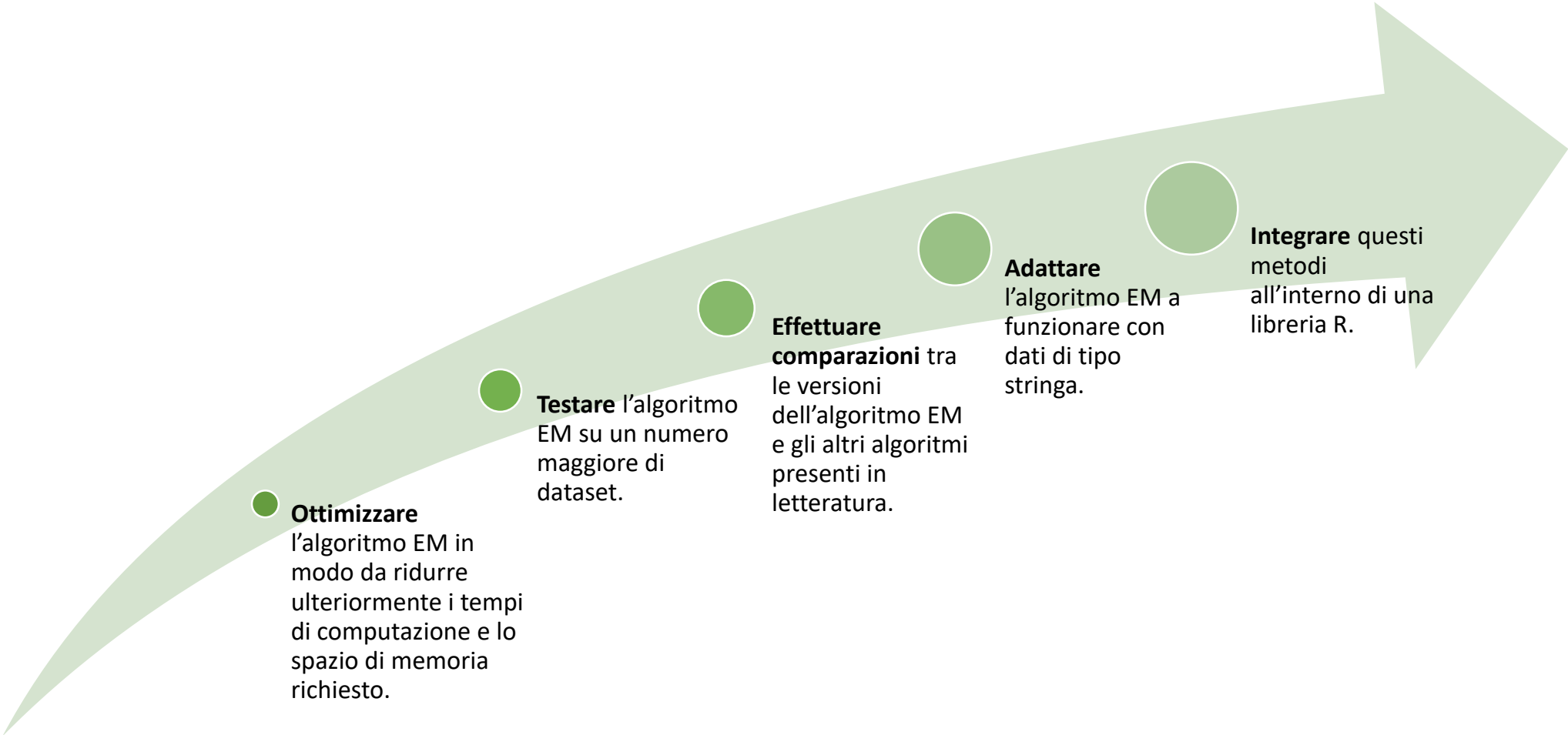
## RISULTATI E CONCLUSIONI

- *EM Hard* **converge più velocemente** e con un numero di iterazioni più basso. Al contrario, *EM Soft* può convergere molto lentamente poiché ad ogni iterazione aggiorna i parametri più cautamente rispetto alla variante *Hard*.
- La scelta dell'iperparametro `NUMBER_ITERATION` risulta essere fondamentale in *EM Soft*. Un numero troppo alto potrebbe condurre a **fenomeni di overfitting**.
- Osservando gli intervalli di confidenza è possibile rimarcare che *EM Hard* fornisce risultati con una deviazione standard minore rispetto alla variante *EM Soft*. **Quando non è facile stabilire quale algoritmo applicare, la scelta consigliata è quella di usare *EM Hard*.**
- *EM HARD* è altresì consigliabile qualora le **capacità hardware sono limitate** e il tempo computazionale risulta essere un problema rilevante.
- Se i dati mancanti sono di tipo **MCAR** valgono, in generale, le **stesse considerazioni** effettuate per i dati mancanti di tipo MAR e MNAR. Tuttavia, il rischio di overfitting per i dati di tipo MCAR e MAR risulta essere maggiore.



# QUALCHE IDEA PER IL FUTURO

SVILUPPI FUTURI



**Ottimizzare**  
l'algoritmo EM in modo da ridurre ulteriormente i tempi di computazione e lo spazio di memoria richiesto.

**Testare** l'algoritmo EM su un numero maggiore di dataset.

**Effettuare comparazioni** tra le versioni dell'algoritmo EM e gli altri algoritmi presenti in letteratura.

**Adattare** l'algoritmo EM a funzionare con dati di tipo stringa.

**Integrare** questi metodi all'interno di una libreria R.