

# Relazione IALab - Uncertainty

Roger Ferrod, Pio Raffaele Fina, Lorenzo Tabasso

Dipartimento di Informatica, Università degli Studi di Torino

`roger.ferrod@edu.unito.it,`

`pio.fina@edu.unito.it,`

`lorenzo.tabasso@edu.unito.it`

## 1 Introduzione

Nella seguente relazione viene illustrato il progetto d'esame per la parte di *reasoning* in presenza di conoscenza incerta. Il documento è diviso in due parti, che si occupano rispettivamente di:

- inferenza esatta su reti bayesiane **statiche**, attraverso l'algoritmo di *Variable Elimination* (VE).
- inferenza esatta su reti bayesiane **dinamiche** attraverso l'algoritmo di *Variable Elimination* (VE) applicato dinamicamente grazie alla tecnica di *Rollup Filtering*.

In questa sezione vengono introdotti brevemente i formalismi ed i concetti teorici, mentre in sezione 2 sono descritti gli esperimenti e i relativi risultati.

*Notazione* Data una rete bayesiana definita su  $\mathbf{X} = \{X_1, \dots, X_n\}$  variabili aleatorie (va) ed una query, individuiamo il seguente partizionamento per le va della rete:

- $\mathbf{Q} = \{Q_1, \dots, Q_p\}$  variabili di query.
- $\mathbf{E} = \{E_1, \dots, E_k\}$  variabili di evidenza.
- $\mathbf{Z} = \mathbf{X} \setminus (\mathbf{Q} \cup \mathbf{E})$  variabili hidden.

Ove presente, il simbolo  $\otimes$  indica che non è stato possibile portare a termine la computazione, per ragioni legate ad errori d'implementazione della libreria AimaCode, oppure per problemi di esaurimento della memoria a disposizione.

### 1.1 Variable Elimination

L'algoritmo VE è uno dei più noti algoritmi di inferenza esatta. Si basa sulla progressiva eliminazione delle variabili, utilizzando operazioni di marginalizzazione sulle variabili hidden.

La complessità temporale e spaziale dell'algoritmo dipende dalla **complessità topologica** della rete bayesiana in input. Nello specifico, la complessità è

esponenziale rispetto alla *treewidth* della rete<sup>1</sup>. Intuitivamente maggiore è la connessione tra i nodi della rete, maggiore è la *treewidth* e di conseguenza maggiore è il tempo richiesto per l'inferenza.

In aggiunta a queste osservazioni, altre importanti analisi sull'algoritmo VE, associano a reti *singly-connected*, come *trees* e *polytrees*, una complessità lineare nel numero di variabili della rete, ovvero  $O(n)$ . Allo stesso tempo, per reti *multiply connected* la complessità è esponenziale nella *treewidth*, ovvero  $O(d^k n)$ , dove  $d = \max_i \text{dom}(X_i)$ , ovvero la massima dimensione del dominio di una variabile aleatoria.

Inoltre, in riferimento all'*induced graph*<sup>2</sup>, la *treewidth* è legata all'ordinamento delle variabili, ovvero all'ordine con cui vengono processate ed eliminate. Più formalmente, la *treewidth* è l'ordinamento che permette di ottenere la minima *width* tra **tutti i possibili ordinamenti**. Ovviamente trovare l'ordinamento migliore tra tutti i possibili  $n!$  ordinamenti è un problema *NP – hard*.

## 1.2 Pruning ed euristiche di ordinamento

I due problemi appena presentati (complessità e ordinamento) possono essere affrontati attraverso tecniche di pruning ed euristiche di ordinamento.

Abbiamo sperimentato tre tecniche di pruning:

- Pruning di nodi irrilevanti (TH1). La tecnica si basa sulla seguente osservazione: siano  $Q$  e  $E$  rispettivamente le v.a. di query ed evidenza, un nodo  $Z$  è irrilevante se  $Z \notin \text{Ancestors}(Q \cup E)$
- Pruning attraverso *m-separation* (TH2). La tecnica verifica l'indipendenza condizionale dei nodi  $Z$  dalla query  $Q$  data l'evidenza  $E$ . Nello specifico viene costruito un *moral-graph* e si ricorre ad un test d'indipendenza tramite *m-separation* per decidere se  $Z$  è irrilevante.
- Pruning di archi irrilevanti (TH3). La tecnica prevede l'eliminazione gli archi  $E_i \rightarrow Z$  tra i nodi di evidenza  $E$  e i nodi  $Z$ . Le CPT dei nodi  $Z$ , a cui sono stati eliminati dei padri, vengono modificate riducendone la dimensione.

Per quanto riguarda le euristiche di ordinamento, abbiamo analizzato:

- *Min-degree*: ad ogni step di eliminazione viene scelta, in modo greedy, la variabile che permette di costruire i fattori più piccoli. Questo corrisponde a selezionare la variabile con il neighborhood minore all'interno dell'*interaction graph*.

<sup>1</sup> Una misura di quanto uno grafo sia strutturalmente simile ad un albero. Tramite la *tree decomposition* è possibile scomporre un grafo in alberi con nodi  $X_1, X_2, \dots, X_n$  dove  $X_i$  è un sottoinsieme dei vertici del grafo originale. La *width* di una *tree decomposition* è la dimensione del più grande nodo dell'albero  $\max_i |X(i)| - 1$ , mentre la *treewidth* è la minima *width* di ogni possibile albero in cui è possibile decomporre il grafo.

<sup>2</sup> Grafo indotto dall'esecuzione dell'algoritmo VE. Ogni eliminazione può essere interpretata come una trasformazione su un grafo; l'unione di queste trasformazioni formano l'*induced graph*.

- *Min-fill-in*: ad ogni step di eliminazione viene scelta, in modo greedy, la variabile la cui eliminazione porta ad aggiungere il minor numero di archi nell'*interaction graph*.

### 1.3 Reti Bayesiane dinamiche

Le reti bayesiane che modellano una sequenza di variabili sono dette reti bayesiane dinamiche. Di conseguenza, oltre alle dipendenze che definiscono una rete statica, nella versione dinamica sono modellate anche le transizioni da un istante temporale ad un altro.

Espandendo la rete dinamica in tutti i suoi istanti temporali (discreti), risulta possibile applicare gli stessi meccanismi di inferenza sviluppati per le reti statiche. Tuttavia, considerato l'elevato costo che questa operazione comporterebbe, l'algoritmo di inferenza è applicato ad una finestra scorrevole, ovvero a *slice* temporali di dimensione ridotta. Tutti gli esperimenti che seguono fanno uso dell'assunzione di Markov del primo ordine, di conseguenza uno stato al tempo  $t$  dipende solamente da altri stati al tempo  $t$  o, eventualmente, da stati al tempo  $t - 1$ .

Più formalmente, data una sequenza di evidenze  $e_{1:t}$  e di variabili di query  $X_{1:t}$ , l'operazione di filtering consiste nel calcolare la distribuzione  $P(X_{1:t}|e_{1:t})$ . Nel caso specifico del *rollup filtering* con assunzione di Markov del primo ordine, ogni slice temporale considera la sottorete negli istanti  $t$  e  $t - 1$ . Dopo aver ridotto la porzione di rete in uno o più fattori, il calcolo procede mantenendo solamente l'informazione associata all'istante temporale precedente.

Il processo di filtering appena descritto può essere effettuato anche ricorrendo a specifici metodi di inferenza approssimata. In particolare è stato utilizzato l'algoritmo *Particle Filtering* (o *Sequential Monte Carlo*) implementato dalla libreria AimaCore.

## 2 Esperimenti e Risultati

Nel pianificare gli esperimenti, sono state individuate quattro *features* principali, listate nella Tabella 1. Considerate in modo congiunto avrebbero comportato l'esecuzione di  $10 \times 5 \times 3 \times 2 = 300$  esperimenti. Per questo motivo si è deciso di condurre 5 esperimenti separati con differenti obiettivi.

Feature	Livelli
Topologia rete	10
Metodo di Pruning	none, th1, th2, th3, all
Ordinamento	topologico, min-fill, min-degree
Nodi di query ed evidenza	avg query, campionamento casuale

**Tabella 1.** Design degli esperimenti.

Per diversificare gli esperimenti, delle 19 reti riportate in Tabella 2, sono state selezionate le reti con valore di *treewidth* differente. A parità di *treewidth* è stata scelta la rete più complessa in termini di *avg. degree* e  $\frac{\text{numero di archi}}{\text{numero di nodi}}$ . Le reti effettivamente utilizzate sono: *earthquake*, *asia*, *sachs*, *win95pts*, *insurance*, *munin* (full), *pigs*, *andes*, *link*.

Network	Nodes	Edges	Treewidth <sup>3</sup>	Treewidth <sup>4</sup>	Max in-degree	Avg. degree
cancer	5	4	1	1	2	1,60
earthquake	5	4	1	1	2	1,60
asia	8	8	2	2	2	2,00
survey	6	6	2	2	2	2,00
child	20	25	3	3	2	2,50
sachs	11	17	3	3	3	3,09
alarm	37	46	3	3	4	2,49
mildew	35	46	4	3	3	2,63
diabetes	413	602	5	4	2	2,92
win95pts	76	112	5	5	7	2,95
pathfinder	109	195	6	5	5	3,58
insurance	27	52	6	6	3	3,85
barley	48	84	6	6	4	3,50
hepar2	70	123	6	6	6	3,51
water	32	66	7	7	5	4,13
munin	1.041	1.397	8	8	3	2,68
pigs	441	592	10	10	2	2,68
andes	220	338	16	15	6	3,07
link	714	1.125	18	16	3	3,15

**Tabella 2.** Reti Bayesiane con le relative statistiche strutturali. Le reti sono pubblicamente disponibili al link <https://www.bnlearn.com/bnrepository/>

Tutti gli esperimenti sono stati eseguiti su una macchina con le seguenti specifiche:

CPU Intel Core i7-7700k (4 core con 2 thread)

RAM 16 GB

Java Heap Space 14GB

## 2.1 AVG Query

Onde evitare l'esplosione combinatoria del numero di esperimenti, dovuta alle diverse query che si possono generare, è stata creata una query che rappresentasse il caso medio di esecuzione dell'algoritmo. In particolare, abbiamo costruito la avg. query seguendo una procedura ben definita basata sulla seguente osservazione di Adnan Darwiche [2]:

<sup>3</sup> utilizzando l'euristica *min-degree*

<sup>4</sup> utilizzando l'euristica *min-fill-in*

*Network pruning can lead to a significant reduction in a Bayesian network if the query variables appear close to the network roots. In the worst case, all leaf nodes appear in the query and, therefore, no pruning is possible. In the best case, the query contains only root nodes which permits the pruning of every node except for those appearing in the query.*

Alla luce di questa osservazione, abbiamo costruito la AVG query come segue:

- Dove possibile la query è del tipo  $P(Q_1, Q_2 | E_1 = e_1, E_2 = e_2, E_3 = e_3)$ . Se la rete non contiene un numero sufficiente di nodi per soddisfare questa condizione, viene scelto il numero di nodi massimo a disposizione (ad esempio, se si hanno solo due nodi disponibili per l'evidenza e non tre come richiesto, verranno selezionati entrambi).
- I nodi di query ed inferenza sono stati selezionati a profondità  $\frac{h}{2}$  della rete.
- La distanza relativa tra i nodi di query ed evidenza non supera (dove possibile) i 3-4 livelli.
- Gli stati delle variabili di evidenza sono stati campionati uniformemente per evitare *selection bias*<sup>5</sup>.

Gli esperimenti  $S0, S1, S2, S3$  sono stati eseguiti sulla avg. query. Alcune visualizzazioni delle avg. query sono illustrate nella Figura 1.

## 2.2 Esperimento S0

L'obiettivo dell'esperimento è la creazione di una **baseline** per poter confrontare i successivi esperimenti. La configurazione degli iper-parametri è la seguente:

- Ordinamento: topologico inverso
- Tecniche di pruning: nessuna

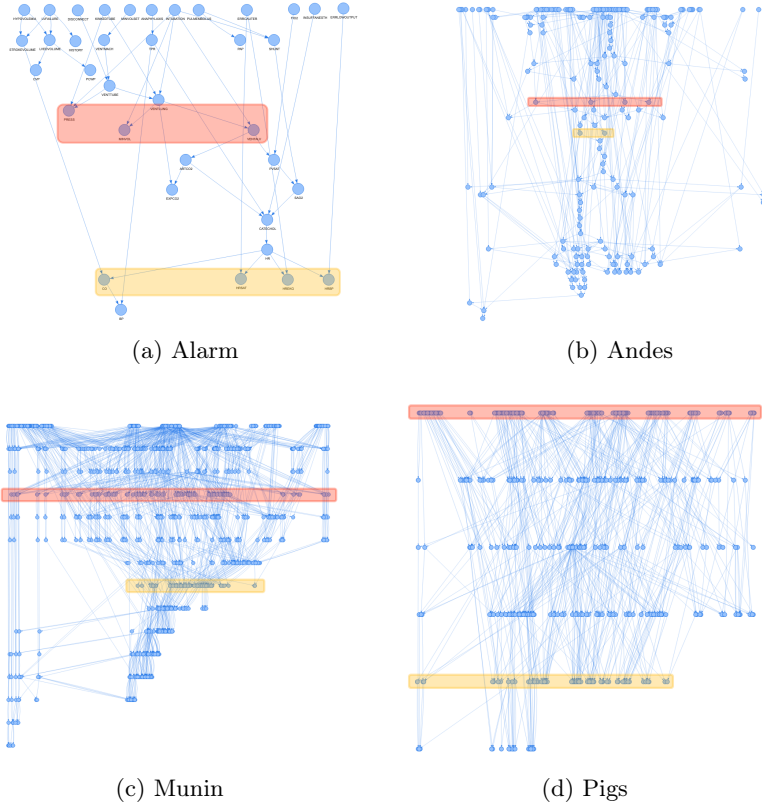
Come si può osservare dalla Tabella 3 il tempo d'esecuzione è correlato alla treewidth. Al crescere della treewidth aumentano infatti le occorrenze del simbolo  $\oslash$ , ossia gli errori dovuti all'eccessiva richiesta di memoria.

## 2.3 Esperimento S1-S2

Gli esperimenti 1 e 2 condividono lo stesso obiettivo: **individuare la migliore combinazione di iper-parametri**. Per evitare di dover valutare tutte le combinazioni di parametri ( $2 \times 4$  esperimenti totali) è stato deciso di condurre i due esperimenti separatamente.

*Esperimento S1* Nel primo esperimento vengono valutati due differenti ordinamenti: *min-fill* e *min-degree*, senza applicare alcuna tecnica di pruning. I risultati sono riportati in Tabella 4. Nelle ultime due colonne della Tabella 4, è possibile

<sup>5</sup> A tal proposito, sono stati effettuati alcuni test in cui si è notato che al variare dello stato scelto, non vi sono significative differenze in termini di tempo d'esecuzione e distribuzione a posteriori.



**Figura 1.** Visualizzaione dell’avg query nelle reti *Alarm*, *Andes*, *Munin* e *Pigs*. I livelli dei nodi di query sono riportati in colore rosso, quelli di evidenza in colore arancione.

apprezzare come i tempi di esecuzione persistano, pressoché inalterati, al variare dell’ordinamento. Inoltre, confrontando i risultati della baseline in Tabella 3, si può notare che le reti associate al simbolo  $\odot$  sono le stesse nella Tabella 4. Questo indica che, in termini di complessità, l’utilizzo di euristiche di ordinamento ha avuto un impatto assai limitato.

*Esperimento S2* Nel secondo esperimento vengono valutate singolarmente le tre tecniche di pruning: TH1, TH2 e TH3. Dato che l’utilizzo delle tecniche non è mutuamente esclusivo, è stata valutata anche una quarta configurazione, che consiste nell’applicazione simultanea delle tre tecniche. Al fine di poter confrontare i risultati con la baseline, è stato utilizzato l’ordinamento topologico inverso. I risultati sono riportati in Tabella 5

Dalla Tabella 5 è possibile osservare come l’ultima configurazione (l’applicazione simultanea di TH1, TH2 e TH3) sia quella che porta a risultati migliori. In questo caso infatti, il calcolo è stato portato a termine nell’80% degli esperimenti,

Network	Treewidth	Width	Time (milliseconds)
earthquake	1	2	2
asia	2	2	2
sachs	3	3	3
alarm	3	5	310
win95pts	5	$\emptyset$	$\emptyset$
insurance	6	4	293
munin_full	8	$\emptyset$	$\emptyset$
pigs	10	$\emptyset$	$\emptyset$
link	15	$\emptyset$	$\emptyset$
andes	16	$\emptyset$	$\emptyset$

**Tabella 3.** Risultati dell'esperimento S0 (baseline); la treewidth è calcolata con l'euristica min-fill.

Network	Treewidth	Width	Time	Min-degree (ms)	Time	Min-fill(ms)
earthquake	1	2	3			3
asia	2	2	4			3
sachs	3	3	28			18
alarm	3	5	78			82
win95pts	5	$\emptyset$	$\emptyset$			$\emptyset$
insurance	6	4	973			970
munin_full	8	$\emptyset$	$\emptyset$			$\emptyset$
pigs	10	$\emptyset$	$\emptyset$			$\emptyset$
link	15	$\emptyset$	$\emptyset$			$\emptyset$
andes	16	$\emptyset$	$\emptyset$			$\emptyset$

**Tabella 4.** Risultati dell'esperimento S1.

con tempi  $< 10$  ms, ovvero +30% rispetto alla baseline. Risulta invece evidente come l'applicazione di una singola tecnica non porti risultati soddisfacenti.

Particolarmente interessante è il caso del TH2. Come già descritto nella sezione introduttiva, l'implementazione utilizzata si basa sul test d'indipendenza condizionale della *m-separation*. Tuttavia, nei casi reali, l'implementazione si basa sulla *d-separation* che può essere eseguita, a differenza della *m-separation*, con complessità lineare [3]. L'utilizzo della *m-separation* potrebbe spiegare l'aumento del numero di  $\emptyset$  nella colonna TH2.

## 2.4 Esperimento S3

L'obiettivo dell'esperimento è effettuare un' **analisi comparativa** tra la **configurazione baseline** e la **configurazione ottima degli iper-parametri**. Alla luce di quanto ottenuto negli esperimenti 1 e 2, abbiamo individuato come configurazione ottima:

- Ordinamento: *minfill*.

Network	Treewidth	TH1		TH2		TH3		All	
		Width	Time (ms)	Width	Time (ms)	Width	Time (ms)	Width	Time (ms)
earthquake	1	2	1	2	1	2	1	2	1
asia	2	2	2	2	2	⊘	⊘	2	2
sachs	3	3	2	3	3	3	3	3	2
alarm	3	4	4	5	310	5	122	4	5
win95pts	5	4	4	⊘	⊘	⊘	⊘	4	4
insurance	6	3	3	4	298	⊘	⊘	3	3
munin_full	8	⊘	⊘	⊘	⊘	⊘	⊘	1	2
pigs	10	3	4	⊘	⊘	⊘	⊘	1	1
link	15	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘
andes	16	⊘	⊘	⊘	⊘	⊘	⊘	⊘	⊘

Tabella 5. Risultati esperimento S2.

- Tecniche di pruning: applicazione simultanea di TH1, TH2, TH3.

I risultati sono riportati in Tabella 6. Da quest’ultima, risulta evidente il **netto miglioramento** nei **risultati** della **configurazione ottima**, sia in termini di tempo d’esecuzione, sia di presenza di errori (⊘).

Nel caso della rete *andes* non è stato comunque possibile portare a termine la computazione. Il problema è stato ulteriormente investigato e, in base all’ipotesi fatta al termine dell’esperimento S2, è stato deciso di escludere il TH2 dalla configurazione ottima e ripetere l’esperimento. I risultati sono riportati in Tabella 7. In questo caso la computazione su *andes* è terminata con successo. Qualitativamente i risultati sono pressoché identici, fatta eccezione per la rete *munin*, che risente dell’assenza del pruning di TH2. Questo risultato non è sorprendente, in quanto tra le 19 reti riportate in Tabella 2, *munin* è la più grande in termini di nodi e connessioni.

Alla luce dei risultati ottenuti sinora, la configurazione ottimale definitiva è quella senza il TH2. Tale configurazione è difatti l’unica che ci permette di terminare la computazione su tutte le reti e, allo stesso tempo, quella in cui le tempistiche di esecuzione non vengono influenzate notevolmente dall’assenza del pruning TH2.

## 2.5 Esperimento S4

L’obiettivo di quest’ultimo esperimento è **valutare l’impatto della tipologia di query sul processo di inferenza**. L’esperimento è stato effettuato scegliendo la configurazione ottimale degli iper-parametri individuata nell’esperimento 3. La rete utilizzata è stata creata tramite il generatore di reti bayesiane *BNGenerator* [1] con i parametri riportati in Figura 2.

Nello specifico, abbiamo effettuato la sperimentazione su tre classi di query:

- Simple query:  $P(Q)$
- Query con evidenza:  $P(Q|E_1, \dots, E_5)$



Network	Treewidth	Width	Time Optimal (ms)	Time Baseline (ms)
earthquake	1	2	2	2
asia	2	2	2	2
sachs	3	3	4	3
alarm	3	4	5	310
win95pts	5	4	5	⊘
insurance	6	3	7	293
munin_full	8	1	3	⊘
pigs	10	1	2	⊘
link	15	4	26	⊘
andes	16	⊘	⊘	⊘

**Tabella 6.** Risultati esperimento S3. L'ultima colonna è stata riportata per comodità di lettura e fa riferimento alla Tabella 3.

Network	Treewidth	Width	Time sub-optimal (ms)	Time Optimal (no TH2) (ms)
earthquake	1	2	2	4
asia	2	2	2	3
sachs	3	3	4	3
alarm	3	4	5	5
win95pts	5	4	5	6
insurance	6	3	7	7
munin_full	8	1	3	33
pigs	10	1	2	6
link	15	4	26	20
andes	16	⊘	⊘	7477

**Tabella 7.** Risultati esperimento S3 senza il pruning TH2.

```
BNGenerator -nNodes 100 -maxDegree 3 -nBNs 1 -fName
test100 -structure singly
```

**Figura 2.** Valore dei parametri d'input a BNGenerator per la generazione controllata della rete.

– Query congiuntiva con evidenza:  $P(Q_1, Q_3, Q_3, |E_1, \dots, E_5)$

Per ogni tipologia, valutiamo l'esecuzione su 10 runs diversi. Per evitare di introdurre *selection bias* durante la costruzione della query, ogni run campiona casualmente (senza ripetizioni) delle variabili di query  $\mathbf{Q}$  e delle variabili di evidenza  $\mathbf{E}$ . In Tabella 8 sono riportati i tempi d'esecuzione ottenuti dai 10 runs.

Tipologia di query	Tempo (ms)	Max width
$P(Q)$	$2,7 \pm 0,66$	2
$P(Q E_1, \dots, E_5)$	$3,7 \pm 0,81$	4
$P(Q_1, Q_2, Q_3,  E_1, \dots, E_5)$	$4,9 \pm 1,20$	3

**Tabella 8.** Risultati esperimento S4. Ogni valore è ottenuto come media aritmetica dei singoli valori nei 10 runs.

Come si evince dalla Tabella 8, all'aumento dei parametri della query e della width, corrisponde un aumento del tempo medio. Questo porta alla conclusione che, una differente struttura della query comporta differenti tempi di esecuzione del processo di inferenza. In modo particolare, ordinando la tipologia di query in base al tempo, possiamo chiaramente vedere che:

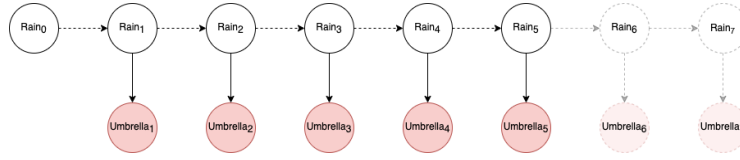
$$simple < evidence < conjunctive$$

## 2.6 Esperimenti Reti Dinamiche

Al fine di implementare una rete bayesiana dinamica, mantenendo la compatibilità con la libreria AimaCode, abbiamo dapprima definito una rete statica (detta "prior") per modellare la distribuzione a priori della rete (istante  $t = 0$ ), successivamente sono stati aggiunti i nodi di evidenza e il *mapping* tra istanti temporali. Visivamente è riportato un esempio di rete bayesiana dinamica in Figura 3.

Sono state create 7 reti dinamiche, mostrate in Figura 4, variando la complessità delle relazioni<sup>6</sup>. Le caratteristiche possono essere riassunte come segue:

<sup>6</sup> Per semplicità, tutte le reti dinamiche hanno dominio booleano



**Figura 3.** La rete *Umbrella* "unrolled" in 7 istanti temporali. Le frecce tratteggiate indicano il padre del nodo all'istante temporale  $t$ , e i nodi di evidenza sono riportati in colore rosso.

- *umbrella* (Figura 4a) presenta un solo stato hidden associato ad un'unica variabile di evidenza; di conseguenza la rete in questione è di fatto una Hidden Markov Model (HMM)
- *wind* (Figura 4b) aggiunge, rispetto alla rete "umbrella", una variabile nascosta, perdendo quindi la proprietà HMM e divenendo una generica rete bayesiana dinamica.
- *twofactors* (Figura 4f) ha la particolarità di generare, durante il processo di inferenza tramite VE, due fattori; la rete è infatti composta da due componenti connessi distinti.
- *earthquake* (Figura 4e) aggiunge, rispetto ai modelli precedenti, la possibilità di avere contemporaneamente collegamenti di diversa natura: il nodo  $C$  infatti possiede sia padri dello stesso istante temporale ( $A$  e  $B$ ), sia nodi di un istante precedente ( $C$ ).
- in *fivestate* (Figura 4c) la complessità della rete aumenta grazie all'introduzione di 4 stati di query e 1 variabile di evidenza.
- la rete *fivestate2* (Figura 4d) rappresenta un'evoluzione di "fivestate" a cui si aggiunge una relazione temporale in un nodo interno ( $C$ ).
- *tenstate* (Figura 4g) rappresenta la rete più complessa, che conta 7 nodi di query e 3 di evidenza, per un totale di 10 stati; le connessioni tra nodi sono varie e considerano sia nodi radice, sia nodi interni.

*Esperimento D0* L'obiettivo del primo esperimento consiste nel **confrontare, senza alcuna euristica, i risultati ottenuti tramite inferenza esatta e inferenza approssimata**<sup>7</sup>. A tal fine è stato calcolato l'errore quadratico medio tra i valori reali (ottenuti dalla nostra implementazione) e i risultati approssimati. Più formalmente, l'errore quadratico medio è definito come:

$$MSE = \frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{n} \quad (1)$$

dove  $x_i$  è il componente  $i$ -esimo della distribuzione reale,  $\hat{x}_i$  è il corrispettivo approssimato e  $n$  la dimensione dei campioni. In questo caso  $n = 2^{|\mathbf{Q}|}$ . I risultati sono riportati in Tabella 9.

<sup>7</sup> L'algoritmo di inferenza approssimata è stato eseguito con un numero di iterazioni pari a 10k

	MSE
umbrella	1,34E-08
umbrellaWind	3,94E-05
twoFactors	1,16E-05
earthquake	1,82E-05
fiveStates	6,13E-06
fiveStates2	1,04E-05
tenstates	9,00E-04
avg. MSE	1,41E-04

**Tabella 9.** Risultati esperimento D0.

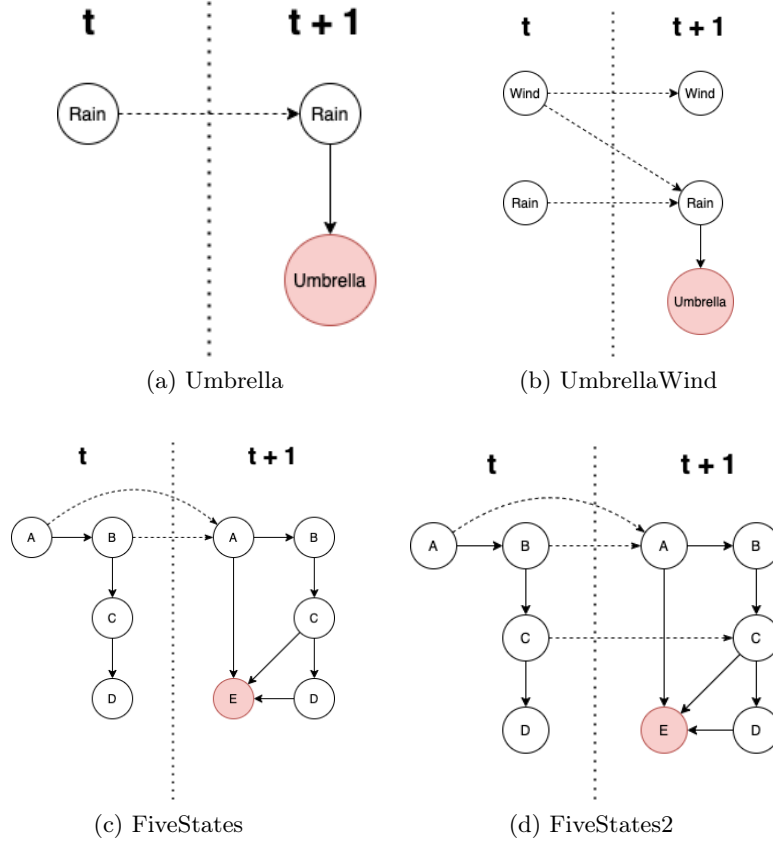
Come si può osservare in Tabella 9, l'errore medio su tutte le reti è  $\approx 1,0E-04$ . Valore, che per gli obiettivi della sperimentazione, si ritiene indicativo di una corretta implementazione della VE con rollup filtering.

*Esperimento D1* L'obiettivo di questo esperimento è **valutare l'impatto delle euristiche relative all'ordinamento di eliminazione**. I risultati sono riportati in Tabella 10. Analogamente a quanto ottenuto nell'esperimento S3 la variazione delle tempistiche ottenute è minima. Questo conferma che le euristiche di ordinamento non hanno un elevato impatto su i tempi richiesti dal processo di inferenza.

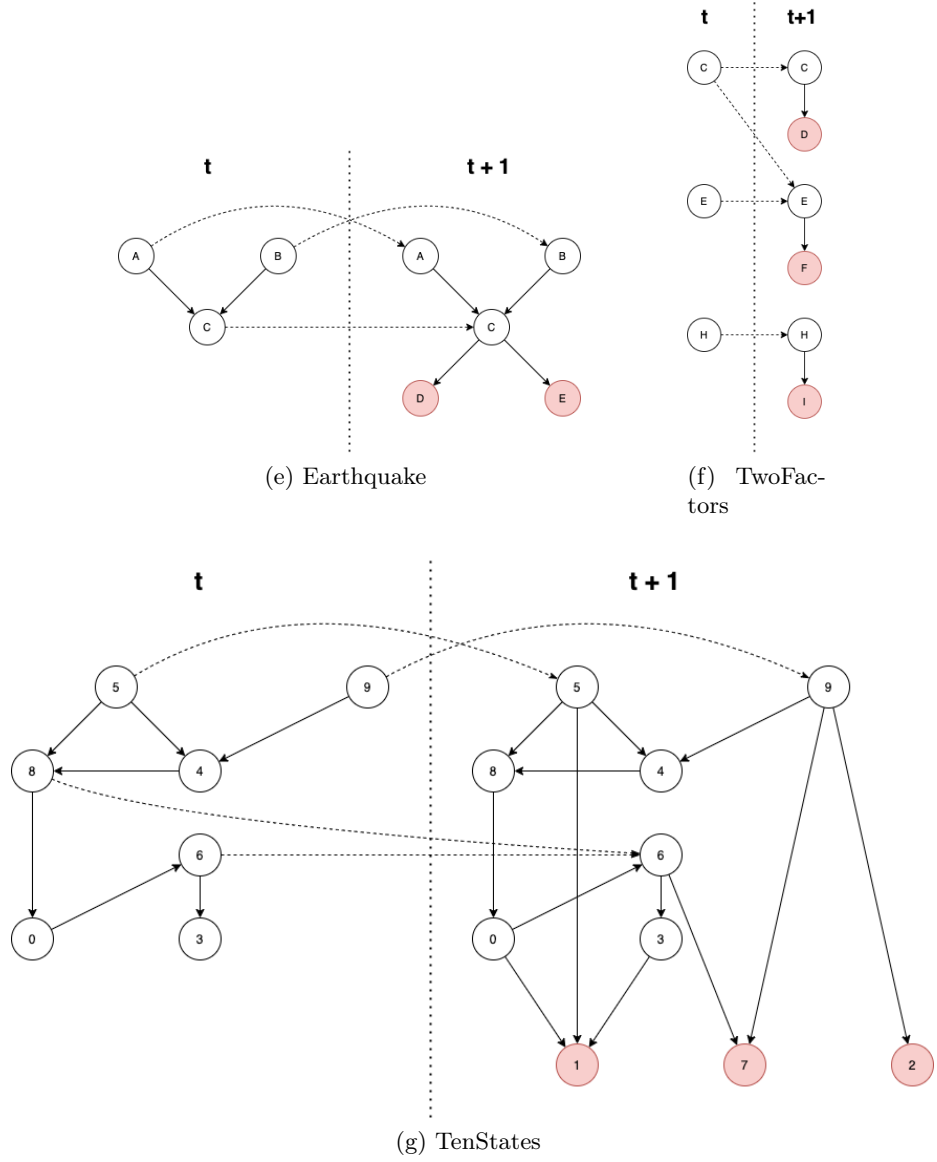
Network	Topological (ms)	Min-degree (ms)	Min-fill (ms)
umbrella	0,49	0,38	0,54
umbrellaWind	0,38	0,5	0,54
twoFactors	0,45	0,44	0,58
earthquake	0,65	0,43	0,56
fiveStates	0,5	0,49	0,47
fiveStates2	0,36	0,4	0,39

**Tabella 10.** Risultati esperimento D1 con i relativi tempi di esecuzione.

## 3 Appendice A



**Figura 4.** Reti bayesiane dinamiche utilizzate negli esperimenti *D0* e *D1*. Le frecce tratteggiate indicano il/i padre/i del nodo al time slice  $t$ , e i nodi di evidenza sono riportati in colore rosso.



**Figura 4.** Reti bayesiane dinamiche utilizzate negli esperimenti *D0* e *D1*. Le frecce tratteggiate indicano il/i padre/i del nodo al time slice  $t$ , e i nodi di evidenza sono riportati in colore rosso.

## Riferimenti bibliografici

- [1] *BNGenerator*. <http://sites.poli.usp.br/pmr/ltd/Software/BNGenerator>. Accessed: 2020-11-09.
- [2] Adnan Darwiche. *Lecture Notes cs673*. [http://web.cs.iastate.edu/~jtian/cs673/cs673\\_spring05/lectures/cs262a-5.pdf](http://web.cs.iastate.edu/~jtian/cs673/cs673_spring05/lectures/cs262a-5.pdf). Accessed: 2020-11-09.
- [3] André Evaristo dos Santos. “Simplifying D-Separation and M-Separation in Bayesian Networks”. English. Tesi di dott. Regina, Saskatchewan: University of Regina, lug. 2016. URL: <https://ourspace.uregina.ca/handle/10294/7635>.