Inferenza in grafi orientati mediante JunctionTree

Semola Rudy (5595074)

Marzo 2018

1 Introduzione

Verso la fine degli anni ′80 Steffen Lauritzen e David Spiegelhalter, esperti statisti, fondarono HUGIN EXPERT oggi un fornitore leader di software di supporto decisionale avanzato che usa l'algoritmo chiamato da molti Hugin per effettuare inferenze probabilistiche. Fornita una strategia implementativa di tale algoritmo di inferenza per grafi orientati basato sul Junction Tree in linguaggio Python, lo scopo di questo elaborato è quello di confrontare i risultati ottenuti con quelli ottenuti da Hugin usando la stessa rete.

2 Rete bayesiana,JunctionTree e l'algoritmo Hugin

L'inferenza probabilistica prevede di calcolare le probabilità di proposizioni a posteriori,poste come query. Una valida rappresentazione grafica delle relazioni di dipendenza assoluta e di dipendenza condizionale tra le variabili di un sistema risulta essere la *rete bayesiana*(DAG).

Tratteremo di un algoritmo esatto di clustering basato sull'idea di compilare il DAG iniziale in un *JunctionTree* (problema NP-difficile); quest'ultimo è un poli-albero con due tipi di nodi,cluster e separatori.Tale costruzione permette ai cluster di propagare l'informazione mediante lo scambio di messaggi. Per fare ciò l'albero dovrà essere inizializzato e garantire la consistenza globale(ovvero verificare la proprietà della run intersection e la consistenza locale).

Il suddetto algoritmo usa lo schema di *propagazione Hugin* che prevede:una fase di inizializzazione, una fase di assorbimento che garantisce la consistenza globale, dopo aver dato evidenza ai nodi del Junction Tree, tale evidenza deve essere propagata agli altri nodi(usando il concetto di vettore finding e del suo teorema associato).

3 Realizzazione del programma per l'inferenza

Il cuore del programma prevede di partire dal JunctionTree e da esso usando l'algoritmo Hugin mostrare come effettuare la propagazione dell'evidenza e determinare attraverso la marginalizzazione la probabilità a posteriori mostrandole su schermo.

3.1 Specifiche e prerequisiti

Python versione 3.5.1: il linguaggio di programmazione scelto per il programma.

pbnt: la libreria usata per l'implementazione della distribuzione delle probabilità condizionali delle variabili della rete,delle tabelle (potenziali) dei nodi del Junction Tree e per determinare da esse la probabilità marginale delle variabili.

HUGIN EDUCATIONAL: celebre software usato per confrontare i risultati ottenuti sulla stessa rete.

Reti Bayesiane: il programma effettua l'inferenza partendo da una semplice rete chiamata *xyz*, prosegue usando una semplice rete presa dal package Hugin Educational chiamata *Fire* ed infine effettua l'inferenza su una rete più complessa creata con l'aiuto del programma Hugin Educational nominata *c*.Il programma mantiene il suo sviluppo: partendo inizialmente su una semplice rete si testava il programma trovando eventuali debug, dopodiché provandolo su reti più articolate si verificava la sua correttezza.

3.2 Documentazione del codice

Il risultato finale prevede 5 moduli: *Variable.py* contiene la classe Variable(una struttura dati per la rappresentazione dei nodi di una rete bayesiana)e quindi il dominio della variabile e la sua cardinalità, la CPD, ed i parenti del nodo; *NodeJT.py* contiene i nodi del JunctionTree, ovvero le classi Clique e Separator contenenti le tabelle (potenziali) dove si effettueranno le propagazioni e i calcoli delle probabilità delle variabili; *Jtree.py* implementa la classe JunctionTree contente il set delle cricche, e le funzioni usate per inizializzare l'albero ed effettuare l'assorbimento; *InferenceEngine.py* contiene le funzioni per costruire i diversi JunctionTree ed è salvata la classe HuginClass(implementa lo speudocodice Hugin ed è l'interfaccia per accedere all'albero, ovvero attraverso questa classe lo rendo inizializzato e consistente, propago l'inferenza e determino le probabilità marginali partendo dalla tabella); *main.py* modulo principale dove si esegue il programma, la sua struttura è stata già spiegata nella precedente sezione parlando delle reti.

4 Esperimenti

Eseguendo più volte il programma,in alcuni casi i valori delle probabilità data evidenza variavano rispetto ai valori di Hugin con un range $0.02 - 0.15$ a seconda della morfologia della rete e in alcuni casi anche dalla scelta della radice per effettuare l'assorbimento. Si è ritenuto interessante quindi tenere traccia di questi fattori durante l'analisi dei risultati sperimentali(ovvero considerare diverse reti e complessità, variare la scelta della radice e effettuare anche più evidenze).

A seguire i diversi risultati confrontati con Hugin e le relative osservazioni.

4.1 Rete XYZ

Iniziamo per gradi;riportiamo la probabilità di una variabile senza evidenza e con l'evidenza in due diverse scelte di nodo radice:

variabile Y	radice	evidenza	P(var)	P Hugin	range
	XY	no	0.2325	0.2325	-
	XY	Z=True	0.8046	0.8046	-

variabile Y	radice	evidenza	P(var)	P Hugin	range
	XZ	no	0.2324	0.2324	-
	XZ	Z=True	0.6644	0.8046	0.1402

Osserviamo su questa semplice rete(*diverging-arrow*) una correlazione pressoché identica ad Hugin tranne per l'ultimo caso, accettabile in situazioni dove l'accuratezza del valore non è molto rilevante.

4.2 Rete Fire del package Hugin Educational

Adesso consideriamo le probabilità calcolate su un gruppo di variabili senza evidenza e con evidenza in due diverse scelte di nodo radice:

variabili	radice	evidenza	P(var)	P Hugin	range
Fire	AFT	no	0.01	0.01	-
Alarm	AFT	no	0.0267	0.0267	-
Tampering	AFT	no	0.02	0.02	-
Fire	AFT	Fire=True	1.0	1.0	-
Alarm	AFT	Fire=True	0.9802	0.9802	-
Tampering	AFT	Fire=True	0.02	0.02	-

variabili	radice	evidenza	P(var)	P Hugin	range
Fire	FS	no	0.01	0.01	-
Alarm	FS	no	0.0267	0.0267	-
Tampering	FS	no	0.02	0.02	-
Fire	FS	Fire=True	1.0	1.0	-
Alarm	FS	Fire=True	0.9802	0.9802	-
Tampering	FS	Fire=True	0.02	0.02	-

Usando una rete del package di Hugin i risultati che emergono sono identici a quelli di Hugin, indipendentemente dalla scelta della radice; facciamo notare che la variabile evidenziata ha probabilità 1.0, come è giusto che sia.

4.3 Rete C creata con il programma Hugin Educational

In conclusione, si prendono tutti i nodi della rete calcolando la probabilità senza evidenza, con evidenza e con più evidenze, su una scelta casuale della radice per vedere i risultati ottenuti:

variabili	radice	evidenza	P(var)	P Hugin	range
c1	C1C2	no	0.5	0.5	-
c2	C1C2	no	0.825	0.825	-
c3	C1C2	no	0.445	0.455	-
c4	C1C2	no	0.89	0.8253	0.0647
c5	C1C2	no	0.8675	0.8675	-
c6	C1C2	no	0.3906	0.3906	-

variabili	radice	evidenza	P(var)	P Hugin	range
c1	C1C2	c3=True	0.0109	0.011	0.0001
c2	C1C2	c3=True	0.7516	0.7516	-
c3	C1C2	c3=True	1.0	1.0	-
c4	C1C2	c3=True	0.8019	0.8003	0.0016
c5	C1C2	c3=True	0.8566	0.8748	-
c6	C1C2	c3=True	0.0199	0.02	0.0001

Notiamo dal calcolo delle probabilità senza evidenza che nella variabile c4 è presente una differenza dell'ordine del 0.06, mentre nelle probabilità con c3 evidenziato a True danno le probabilità su c4 che differiscono di molto meno e più variabili hanno differenza di valori dell'ordine minore dei 0.001.

Proviamo adesso a fare un'ulteriore evidenza su una variabile stavolta con valore False,ovvero $P(var|c3 = True, c5 = False)$:

variabili	Probabilità	P Hugin	range
c1	0.0	0.0123	0.0123
c2	0.9769	0.9008	0.0761
c3	1.0	1.0	-
c4	0.80	0.8004	0.0004
c5	0.0	0.0	-
c6	0.0199	0.02	0.0001

Le variabili con probabilità che differiscono da quelle di Hugin aumentano anche se di piccoli valori (ad esclusione della variabile c2 con range dell'ordine dei 0.07).

Facciamo presente senza mostrare la tabella che scegliendo un'altra radice, per esempio C1C3, la precisione è molto superiore e il numero di variabili con valori non correlati diminuisce e due.

5 Conclusioni

Basandosi sui risultati ottenuti il confronto di questi due programmi non mostra risultati molto diversi potendo conclude che,in prima approssimazione,il programma creato funzioni correttamente. Dobbiamo comunque considerare che HUGIN EDUCATIONAL utilizza algoritmi molto più sofisticati e che le piccole variazioni ottenute possono essere frutto di molteplici fattori non considerati come le eventuali approssimazioni vicino a 0 e a 1 che il software di fama mondiale adotta per rendere i risultati sempre più verosimili, il modo effettivo in cui le tabelle vengono create, calcolate e il grado di approssimazione dei suoi valori.