

README

COME RIPRODURRE I RISULTATI

L'elaborato è stato eseguito su Jupyter Notebook.

Per riprodurre i risultati sottoposti è sufficiente:

- Installare le librerie pgmpy e graphviz
- Inserire nella sezione 3 del codice il nome del file (*asia.bif* o *cancer.bif*)

Le reti *asia.bif* o *cancer.bif* su cui è stato testato l'algoritmo sono reperibili su

<https://github.com/ncullen93/pyBN/tree/master/data>

PARTI DELL'ELABORATO NON SVILUPPATE IN AUTONOMIA

- Per il campionamento della rete è stato usato il metodo *forward_sample()* della libreria *pgmpy*.

SPIEGAZIONE DEL CODICE

Il codice è diviso in 8 sezioni.

Nella prima sezione vengono definite le classi *Parameter*, *Node* e *BayesianNetwork*. All'interno della classe *Node* viene definita il metodo *get_config* che, dati il livello del nodo e la configurazione dei suoi padri, restituisce il numero di configurazione. Questo consente di individuare uno specifico parametro data la configurazione del nodo. Tale configurazione è calcolata sfruttando la codifica posizionale. Le configurazioni vanno da 0 a *num_config-1* dove nella prima configurazione il nodo e i suoi padri hanno livello pari a *levels[0]* (dove *levels* è la lista associata a ciascun nodo che contiene tutti i suoi possibili livelli), mentre nell'ultima hanno livello pari all'ultimo elemento nella lista *levels*. Il numero di livello del nodo (compreso tra 0 e *len(levels)-1*) rappresenta la cifra più significativa, mentre il peso dei padri dipende dalla loro posizione nella lista *parents* (il nodo in *parents[0]* ha peso minore) e viene calcolato dal metodo *get_factor*.

Nella seconda sezione si definisce la funzione *BayesianLearning* per l'apprendimento dei parametri da un dataset utilizzando la strategia descritta in Heckerman 1977.

Nella terza sezione si legge il file *.bif* in cui è memorizzata la rete e si crea la rete bayesiana.

Nella quarta sezione si rappresenta graficamente la rete sfruttando la libreria *graphviz*.

Nella quinta sezione si crea la rete bayesiana di riferimento su cui effettuare il confronto.

Nella sesta sezione si definisce la funzione per il calcolo della divergenza di Jensen-Shannon. Essa sfrutta il metodo *get_prob_list* della classe *BayesianNetwork*. Questo metodo ricorsivo permette di calcolare la probabilità di ciascuna configurazione della rete e inserirla in una lista.

Nella settima sezione si apprende i parametri da dataset di dimensione crescente e, per ognuno di essi, si calcola la divergenza di JS. Per garantire un comportamento monotono della curva si creano più dataset di uguale dimensione e si calcola la divergenza media. Naturalmente dataset di piccola dimensione comportano una maggiore variabilità dei risultati e ne viene quindi generato un numero maggiore rispetto a quelli di grande dimensione.

Nella ottava sezione stampiamo le probabilità apprese dall'ultimo dataset della sezione precedente.