Apprendimento di parametri in modelli grafici

Alberto Baldrati

9 febbraio 2019

1 Presentazione del progetto

In questo progetto è inizialmente stato scritto del codice per l'apprendimento di parametri in rete orientate utlizzando l'approccio a massima verosimiglianza descritto in Russel & Norvig 20.2.1.
Successivamente, tramite il software *Hugin Educational*, si sono generati dataset di esempi a partire da una distribuzione nota p e tramite il codice scritto in precedenza i parametri sono stati appresi dalla rete.
Per evitare stime degeneri si sono usati come priors pseudo-counts unitari, ovvero è stato utilizzato Laplace smoothing.
Infine è stata misurata la "distanza" tra la distribuzione appresa su un set di dimensione n, q_n e la distribuzione p tramite la divergenza di Kullback-Leibler definita come:

$$KL(p||q_n) = \sum_U p(U) \log \frac{p(U)}{q_n(U)}$$

È bene notare come nel caso in cui $p(x) = 0$ il termine della disuguaglianza è interpretato come zero, infatti $\lim_{x \to 0^+} x \log x = 0$

2 Descrizione del codice

Il codice da me sviluppato è diviso in 5 file, che possiamo suddividere in tre categorie:

- I file **BayesianNetwork.py** e **Node.py** definiscono la struttura della rete bayesiana.
- I file **Fire.py** e **ChestClinic.py** definscono due istanze concrete di reti bayesiane.
- Il file **Test.py** come suggersice il nome serve ad automatizzare i test su dataset di dimensioni differenti.

2.1 BayesianNetwork e Node

Come accennato in precedenza questi due file servono a definire la struttura della rete bayesiana.
La classe **Node**, come suggerisce il nome, rappresenta un nodo della mia rete bayesiana, essa contiene al suo interno tutto il necessario per poter apprendere informazioni da un dataset, in particolare i metodi più interessanti sono:

- *build_occorrences_dictionary* costrusice il dizionario dove poi verranno salvate le occorrenze durante l'apprendimento. Per fare ciò utlizza la lista dei padri e i relativi domini di essi (ovvero i loro *domain_values*).
  Il dizionario è inizializzato a 1 per evitare stime degeneri.

- *add_occorences_to_occ_dictionary* aggiorna il dizionario delle occorrenze a seconda dei valori che gli vengono passati in input.

- *build_probability_dictionary* date le occorrenze costruisce il dizionario delle probabilità, il quale verrà successivamente confrontato con la distribuzione nota *p*.

La classe **BayesianNetwork** è una classe astratta che definisce un "interfaccia" per le reti concrete, le quali dovranno estendere tale classe ed implementare i suoi metodi astratti.
i metodi principali di questa classe sono:

- *graph_build_occorrences* si preoccupa di richiamare su tutta la rete il metodo della classe Nodo *graph_build_occorrences*.

- *graph_count_occorrences* si preoccupa di leggere il file di input (che va speficiato al momento della creazione della rete concreta) e di aggioranare i contatori di occorrenze presenti in ogni nodo.

- *graph_build_probabilities* si preoccupa di richiamare su tutto il grafo il metodo della classe Nodo *build_probability_dictionary*.

- *calculate_Kullback_Leibler_distance* ritorna il valore della divergenza di Kullback-Leibler dispetto alla distribuzione nota e quella appresa.

2.2 Fire e ChestClinic

Entrambi i file contengono le omonime classi.
Sia la classe Fire sia ChestClinic ereditano dalla classe BayesianNetowork implementando i metodi astratti presenti in essa, ovvero:

- *build_graph* definsice la struttura del grafo, creando nodi ed archi.

- *build_real_distribution* definisce i valori della distribuzione nota *p*.

2.3 Test

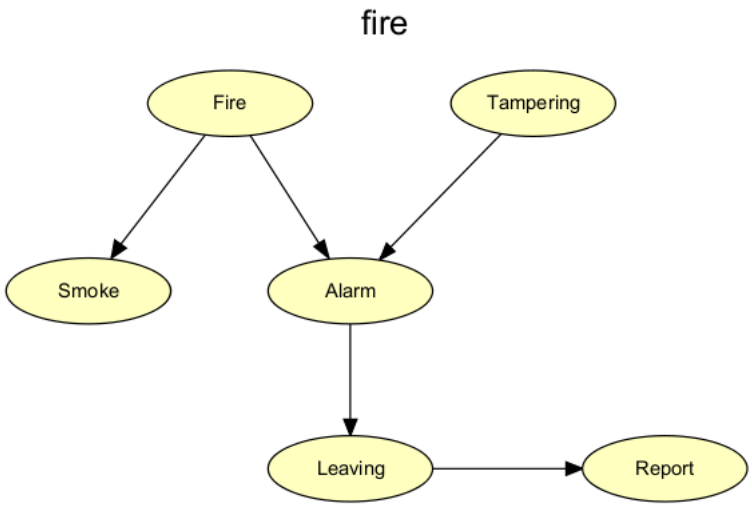
Nel file Test sono presenti i metodi che permettono di appurare il corretto funzionamento dell'apprendimento.
Di default i test sono realizzati sulle due reti utlizzando i dataset presenti nei file *fire.dat* e *chestclinic.dat* (entrambi i file sono presenti nel progetto su github).
I test sono stati ripetuti su un numero di occorrenze differenti per verificare che all'aumentare del numero di esse la divergenza diventi via via minore.

3 Risultati sperimentali

Di seguito verrano descritti i problemi e verranno esposti i risultati sperimentali

3.1 Fire network

3.1.1 Descrizione della rete



NB: I valori della distribuzione p per questa rete bayesiana si possono consultare nel file *firetables.pdf* oltre che direttamente dal file *fire.net* mediante il software Hugin

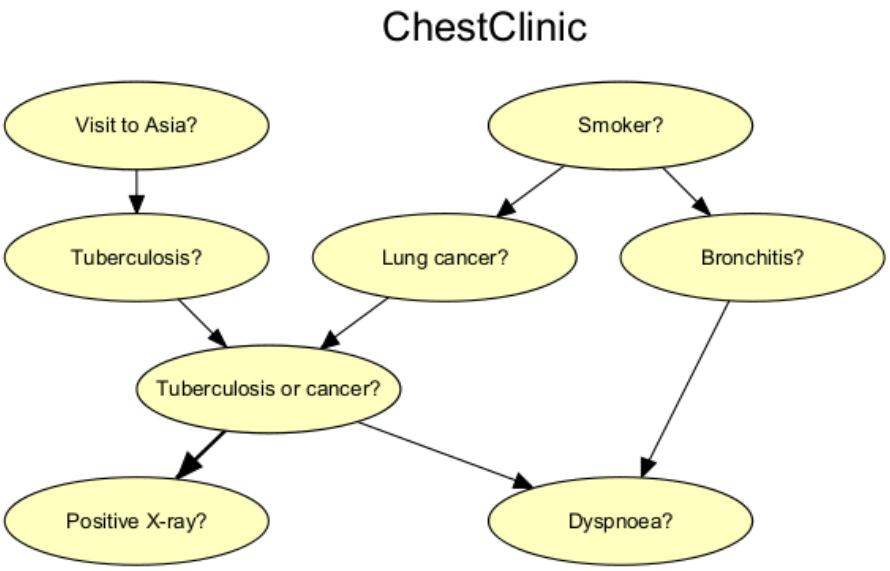
3.1.2 Risultati sperimentali

Divergenza Kullback-Leibler	
Dimensioni dataset input	KL-distance
100	2.73580
1000	0.19164
10 000	0.22867
100 000	0.01841
1 000 000	0.00064

Dai risultati in tabella possiamo notare come all'aumentare delle dimensioni del dataset la divergenza tenda a diminuire arrivando a valori vicini a zero.
Tuttavia notiamo come questo fatto non sia certo, ma solo molto probabile, infatti passando da 1000 a 10 000 campioni notiamo un aumento della divergenza.

3.2 ChestClinic Network

3.2.1 Descrizione della rete



NB: I valori della distribuzione p per questa rete bayesiana si possono consultare nel file *chestclinictables.pdf* oltre che direttamente dal file *ChestClinic.net* mediante il software Hugin

3.2.2 Risultati sperimentali

Divergenza Kullback-Leibler	
Dimensioni dataset input	KL-distance
100	2.72292
1000	1.34115
10 000	0.35297
100 000	0.02507
1 000 000	0.00284

Anche in questo caso notiamo una riduzione della divergenza all'aumentare del dataset, arrivando anche qua a valori molto vicini allo zero.
A differenza di prima tuttavia questo volta non abbiamo "anomalie", ovvero aumenti della divergenza all'aumetare dei dati campionati dalla rete.