# Applicazione di una Rete Bayesiana alle recensioni di TripAdvisor

Simone Bertini - Matr. 806867
Luca Moretti - Matr. 807613
Riccardo Nigrelli - Matr. 846986

Repository

7 luglio 2019

## 1 Introduzione

Il progetto consiste nel provare a predire la valutazione di una recensione di un hotel fatta da un utente analizzando la parte testuale della stessa. Nel dettaglio, il progetto lo possiamo strutturare nei seguenti punti:

- rappresentare in modo vettoriale le recensioni del dataset;
- svolgere la fase di selezione delle features;
- modella il problema utilizzando una rete bayesiana;
- effettuare o indurre una stima dei parametri e delle CPT;
- stimare le performance del modello;
- verificare il corretto funzionamento del modello attraverso una demo.

## 2 Analisi del dataset

Il dataset fornitoci era già stato suddiviso in Training e Testing. Tutti i file era scritti in pseudo markup e contengono un insieme di recensioni riguardanti il medesimo hotel.

Tutti i file hanno il seguente header:

```
<Overall Rating>
<Avg. Price>
<URL>
```

Figura 1: Header delle recensioni

i cui elementi rappresentano rispettivamente la media del punteggio totale di tuttle le recensioni, il costo medio e l'indirizzo di TripAdvisor dell'hotel. e la struttura delle recensioni è la seguente: Come si può notare dalla figura 2,

```
<Author>
<Content>
<Date>
<img>
<No. Reader>
<No. Helpful>
<Overall>
<Value>
<Rooms>
<Location>
<Cleanliness>
<Check in / front desk>
<Service>
<Business service>
```

Figura 2: Struttura della recensione

i campi che costituiscono una review sono: il nome dell'autore, la data e il contenuto della recensione, un'immagine (opzionale), il numero di persone che hanno letto la review e il numero di quelli che l'hanno trovata utile e le valutazioni delle seguenti *features*: OVERALL, VALUE, ROOMS, LOCATION, CLEANLINESS, CHECK IN / FRONT DESK, SERVICE, BUSINESS SERVICE.
Tra questi elementi presenti all'interno di una recensione, quelli che andremo ad utilizzare oltre al contenuto della recensione, sono alcune di quelle che nel paragrafo precedente erano state denominate *features*.

# 3 Selezione delle features

Inizialmente abbiamo dovuto decidere quali parole andavo bene per poi rappresentare una recensione e per fare abbiamo eseguito una fase di preproces-

sing del testo.
Per prima cosa, dopo aver letto tutti i testi delle recensioni li abbiamo scritti in un unico file CSV e successivamente ad ogni singola review è stata sottoposta a questa serie di operazioni:

1. conversione del testo in minuscolo;
2. rimozioni di tutte le forme contratte, ad esempio YOU'LL viene convertito in YOU WILL;
3. il test viene "spezzato" in parole singole, le quali rappresentano i TOKEN;
4. rimozione delle STOPWORDS
5. infine, rimozione di tutti i simboli di punteggiatura.

Alla fine di questa fase, si è deciso di prendere i termini che venivano utilizzati con maggiore frequenza i quali possono influire maggiormente sull'OVERALL. Le parole che sono state scelte sono:

| Parola | Frequenza |
|---|---|
| great | 143.473 |
| good | 110.660 |
| comfortable | 27.861 |
| clean | 48.595 |
| bad | 18.381 |
| old | 16.955 |

Tabella 1: Parole selezionate con relativa frequenza

# 4 Rappresentazione dei vettori

Per costruire i vettori, sono stati presi in considerazione tutti i file di Training, tenendo presente che un lavoro simmetrico dovrà essere applicato in seguito ai file di Test. Sono stati scelti dei termini particolari da inserire nel vettore. L'individuazione di questi termini è stata descritta nel paragrafo precedente. Riassumendo, i termini scelti sono: great, good, clean, comfortable, bad e old.
Per ogni review è stato costruito un vettore numerico. Un vettore è composto da 12 parametri. I parametri sono quelli descritti precedentemente, ovvero

la presenza o meno di uno dei 6 termini sopra citati e i 6 Meta-Dati scelti descritti nel paragrafo 2. Per quanto riguarda i termini, viene indicato un 1 quando il termine corrispondente ‘e presente almeno una volta nel testo della recensione, uno 0 altrimenti. Per quanto riguarda gli score, viene riportato esattamente il punteggio inserito dall'utente, questo varia da -1 a 5 inclusi. I vettori sono stati inseriti in un file in formato csv, ottenendo 191.409 righe. Dato che, probabilmente, il valore -1 corrisponde ad un missing value, è stato deciso di rimuovere i vettori contenenti almeno un punteggio -1 in corrispondenza dei metadati. Il file risultante da questo filtraggio contiene esattamente 94.913 vettori.
I vettori sono stati generati utilizzando uno script python che è possibile visionare nella repository GitHub.

# 5 Scelte di modellazione

Per modellare la rete, è stato deciso che i termini che abbiamo selezionato nel corso dell'analisi della parte testuale influenzino il valore di Overall, mentre quest'ultimo può essere visto come il sentiment dell'utente, che quindi influenza i metadati che rappresentano a macro aree le caratteristiche dell'hotel.
Come è possibile vedere in figura 3, l'informazione si propaga dalla presenza o meno dei termini che abbiamo scelto come feature verso Overall, che esprime il sentiment generale sul soggiorno in hotel dell'utente. Il sentiment, quindi, influisce sulla valutazione dell'esperienza dell'utente, andando a caratterizzare i metadati. Infatti, in presenza di un Overall positivo, ad esempio 5, è possibile aspettarsi che l'utente esprima il suo sentiment con aggettivi come *great*, *good* o *clean*. Allo stesso modo, la presenza di un sentiment negativo, come ad esempio Overall 1, è possibile porti all'utilizzo di aggettivi negativi come *bad*, *small* o *old*.
Le scelte di modellazione hanno seguito un ragionamento Knowledge-Based, pertanto, potrebbero non essere le migliori e verranno valutate in seguito, confrontando la struttura descritta con una autogenerata.

## 5.1 Probabilità e CPT

Una volta definita la struttura della rete, sono state stimate le CPT del nostro modello tramite libreria *libpgm* ed il risultato è stato memorizzato in un file json. Per svolgere il task di stima delle CPT è stato necessario utilizzare come parametri la struttura della rete e i dati riguardanti le reviews in formato vettorizzato.

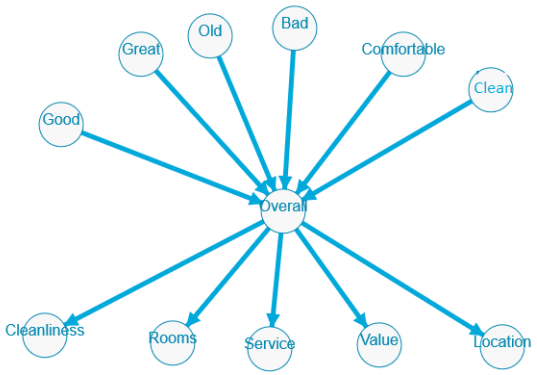


Figura 3: Rete Knowledge-Based

# 6 Confronto con il modello indotto

Come specificato nel precedente paragrafo, al fine di effettuare un confronto tra due modelli di rete Bayesiana differenti, è stato scelto di realizzare un secondo modello indotto a partire dai dati a nostra disposizione. La struttura della rete è visibile in figura 4. Per fare questo è stato utilizzato un metodo della libreria libpgm. Il metodo in questione è DISCRETE ESTIMATEBN al quale è possibile passare come parametro dei dati e ottenere una rete Bayesiana completamente appresa autonomamente. La rete bayesiana ottenuta in questo caso è stata la seguente: Essendo che l'accuratezza della rete non era soddisfacente (in quanto era solo del 46%) abbiamo effettuato delle piccole modifiche alla struttura della rete.
Le modifiche effettuate sono state di tipo Knowledge based e sono state costituite da:

- rimozione di alcuni archi considerati da noi errati o inutili;
- inserimento di alcuni Archi Soprattutto entranti al nodo OVERALL che sarà poi il nodo query per il nostro progetto

Questo modifiche hanno aumentato l'accuratezza globale del modello portandola a un risultato del 71%.

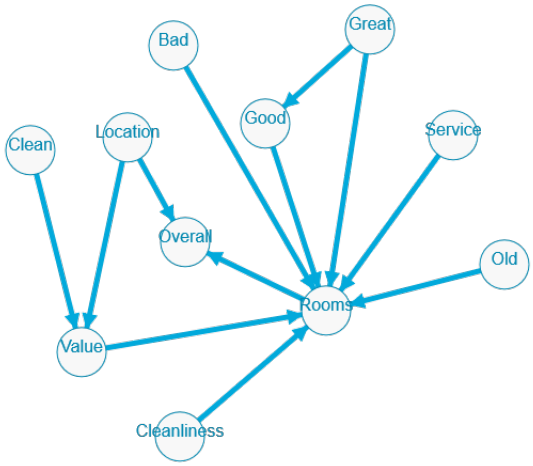


Figura 4: Rete indotta

### 6.1 Probabilità e CPT indotte

Per definire le CPT di questo modello è stata seguita la stessa procedura descritta nel paragrafo 5.1. Per confrontare le CPT con quelle generate dal modello definito da noi, è possibile visualizzare il file json relativo nel nostro repository github.

## 7 Esperimenti

### 7.1 Rappresentazione vettoriale

Procedendo in modo simmetrico rispetto a quanto descritto nel paragrafo 4, sono stati utilizzati tutti i file presenti nella cartella Testing. Il file csv risultate ha subito poi un filtraggio per rimuovere valori non validi (sono stati rimosse le righe con metadati con valore di "-1").

### 7.2 Training del modello

Sia la rete Bayesiana Knowledge-based sia quella auto generata saranno utilizzate per caricare il modello di apprendimento utilizzando la libreria *libpgm*.

Il passo successivo ora sarà testare il modello ottenuto con i vettori di test generati nel sotto paragrafo precedente.

### 7.3 Inferenza

Per testare il modello è stato eseguito un task di inferenza esatta, il quale richiede una variabile Query e alcuni valori osservati, definiti come evidenze. Si è scelto di tenere come evidenze, tutti i valori nel singolo vettore ad eccezione del valore di Overall, utilizzato come variabile Query. In particolare, è stato deciso di valutare il modello con tutti i vettori presenti nel Test e contare tutte le volte che il modello classifica nel modo corretto rispetto a tutte le volte in cui sbaglia.

## 8 Misure di performance del modello

Dopo aver effettuato il task di inferenza su tutti i dataset di Testing siamo andati ad analizzare la correttezza delle previsioni in base all'accuratezza. L'ACCURATEZZA, viene definita come il rapporto della la somma del numero di true positive e true negative con la somma di true positive, true negative, false positive e false negative:

$$\text{Accuratezza} = \frac{tp + tn}{tp + tn + fp + fn}$$

Le performance attenute dai due modelli presi in esame sono rappresentati in tabella 2.

| Classe | Modello Knowledge-Based | Modello Indotto |
|---|---|---|
| 1 | 83% | 68% |
| 2 | 62% | 36% |
| 3 | 61% | 20% |
| 4 | 66% | 73% |
| 5 | 85% | 73% |

Tabella 2: Confronto dell'accuratezza

Con un risultato complessivo pari a 83% per il modello Knowledge-Based e 71% per il modello indotto.

# 9 Analisi Risultati

La misura di accuratezza complessiva tiene conto del fatto che i dati di test siano molto sbilanciati verso le classi 4 e 5 che insieme coprivano più del 75% dei casi testati.
Possiamo motivare il fatto che il nostro modello abbia risultati migliori rispetto al modello indotto in quanto il nostro si focalizza sull'overall mentre quello indotto cerca di costruire una struttura più "generica".

# 10 Conclusioni

Le performance potrebbero salire ulteriormente se passassimo da un problema multiclasse a un problema binario (recensione positiva/negativa).
Basandoci sulle performance possiamo però dire che il nostro modello ha dei buoni risultati.
Dopo aver fatto molti esperimenti sul modello ci siamo resi conto che le parole influenzino molto meno il modello rispetto ai metadati; la presenza o mancanza di keyword modifica le percentali di distribuzione di probabilità nell'ordine del 4/5%.