Stima dell'interesse per topic editoriali ispirata all'Item Response Theory

Alberto Arletti

8 aprile 2021

Neosperience s.p.a.

Introduzione

Un’azienda del settore Energy ha un proprio wito web un blog con più di 500 articoli su tematiche differenti: ambiente, tecnologia o risparmio.

È interessata a conoscere a quale di questi topic ogni utente è più interessato.

I dati a nostra disposizione rappresentano il tempo (attivo, di lettura) trascorso da ogni cliente su ciascuna pagina.

Dataset

	pagina 1	pagina 2	pagina 3	...	pagina m
utente 1	20	0	0	...	0
utente 2	0	4	0	...	0
...	...	...	...	...	2
utente n	0	1	0	...	0

Il dataset quindi consiste in una matrice abbastanza sparsa in cui le colonne indicano le pagine web, le righe gli utenti e i valori rappresentano il tempo di lettura di ogni pagina da quell'utente.
Il numero di utenti è 1736, il numero di pagine 553.

Un dizionario fornito dall'azienda permette di associare ogni pagina al proprio topic. Alcune pagine possono appartenere a più di un topic.

Obiettivo e sfide

La richiesta è quella di stimare l'interesse di ogni utente per ciascun topic.

Sfide del problema:

- Non tutte le pagine hanno la stessa probabilità di essere lette: alcune pagine sono poste in evidenza in homepage, altre sono più nascoste.
- I clienti sono molto diversi fra loro: alcuni entrano solo una volta sul blog e poi spariscono altri leggono molte pagine nei mesi.
- Non esiste un *groud truth*: non abbiamo modo di chiedere agli utenti il loro interesse esplicitamente (o di verificare l'accuratezza della stima attraverso un recommendation system)

Assunzioni e ipotesi

Diamo per assunto che:

- Le pagine trattino ciascun topic in modo binario (o ne parlano o non ne parlano). Questa è una semplificazione poiché in realtà alcune pagine parlano più estensivamente di un topic mentre altre pagine lo trattano solo superficialmente.

Una nostra ipotesi è che:

- Semplicemente leggere più pagine di uno stesso topic non corrisponda automaticamente ad un maggior interesse per quel topic.

Approccio

Per affrontare il problema è stato sviluppato un approccio ispirato alla Psicometria: la Item Response Theory (IRT)

L'approccio della
IRT è quello di cercare di stimare sia l'abilità di chi compie il test, sia la difficoltà di ogni domanda del test. Più abile è il soggetto, maggiore sarà la sua probabilità di rispondere a domande più difficili.

Approccio

Nel nostro caso d'uso, ispirandoci all'Item Response Theory (IRT), possiamo considerare le pagine come domande o items di un test, e ogni cliente come un soggetto del test.

L'abilità del cliente quindi rappresenterà il suo interesse verso un determinato topic. Maggiore sarà il suo interesse, maggiore sarà la proabilità di visitare pagine che richiedono un alto interesse per essere lette.

Implementazione

Per dati dicotomici, la formula di Rasch è espressa come:

$$P\{X_{ni} = 1\} = \left(\frac{e^{(\alpha - \gamma)}}{1 + e^{(\alpha - \gamma)}} \right)$$

Dove α rappresenta l'abilità e γ rappresenta la difficoltà.

Operativamente, per stimare α e γ si può fare risolvere l'equazione:

$$\log P(X|\alpha, \gamma) = \sum_{q,s} x_{qs} \log \left(\frac{e^{(\alpha - \gamma)}}{1 + e^{(\alpha - \gamma)}} \right) + (1 - x_{qs}) \log \left(1 - \left(\frac{e^{(\alpha - \gamma)}}{1 + e^{(\alpha - \gamma)}} \right) \right)$$

dove l'obbiettivo è trovare dei parametri α e γ che massimizzino la probabilità di osservare i nostri dati.

Definizione dei priors (Python)

Costituiamo dei priors indipedenti per α e γ tramite una distribuzione normale con media 0 e deviazione standard 1.

```
alpha = pm.Normal('Person', mu = 0, sigma = 1,
                  shape = (1, len(data)))
gamma = pm.Normal('Question', mu = 0, sigma = 1,
                  shape = (data.shape[1], 1))
```

Stima dei Likelihood (Python)

La distribuzione del likelihood viene stimata con l'equazione prima mostrata.

```
def logp(d):
    v1 = tt.transpose(d) * tt.log(tt.nnet.sigmoid(
        tt.exp(alpha - gamma) /
        (1 + tt.exp(alpha - gamma))
    ))

    v2 = tt.transpose(
        (1-d)) * tt.log(1 - tt.nnet.sigmoid(
        tt.exp(alpha - gamma) /
        (1 + tt.exp(alpha - gamma))
    ))

    return v1 + v2

ll = pm.DensityDist('ll', logp,
        observed = {'d': data.values})
```

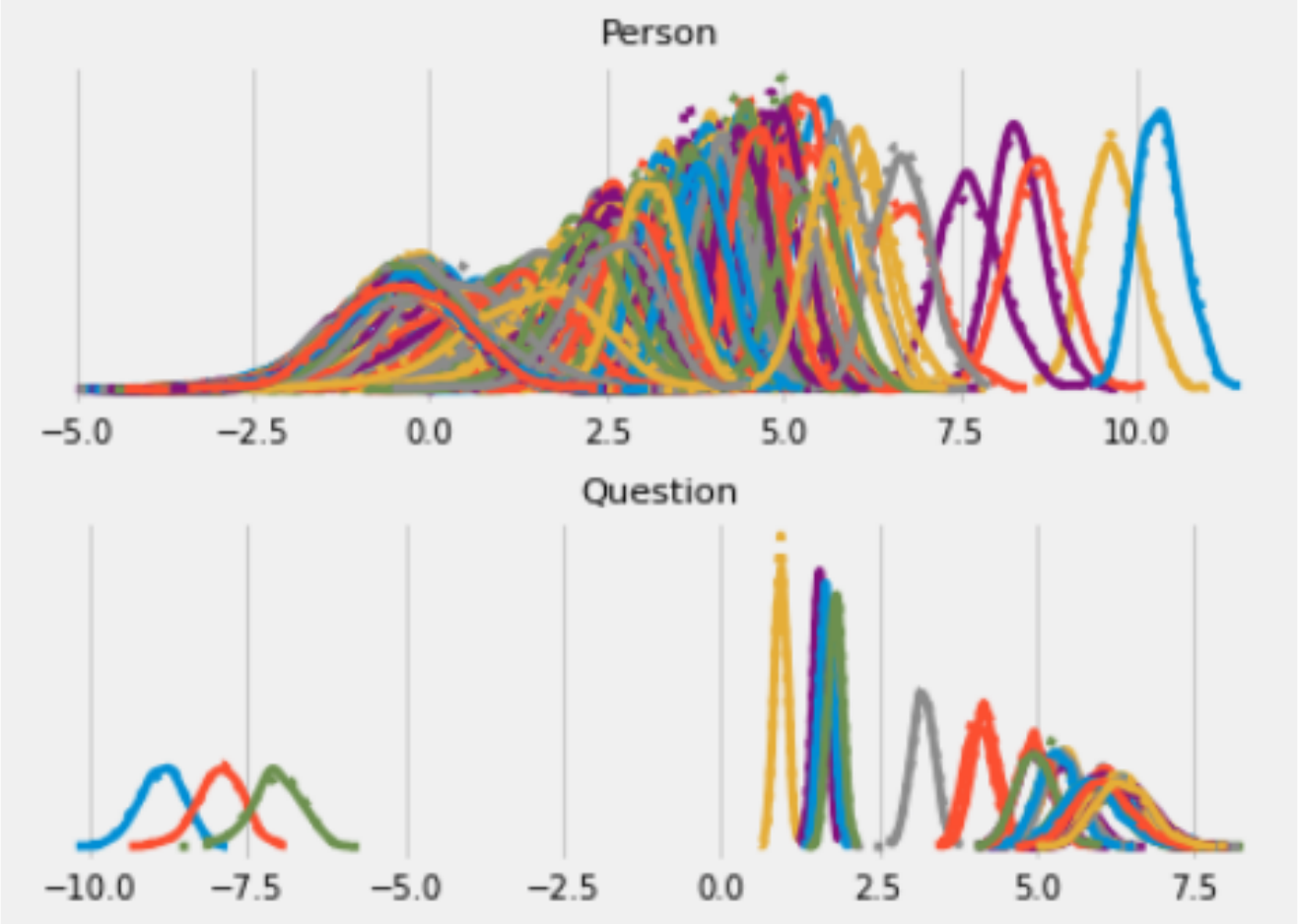
Campionamento Posterior (Python)

I posteriors vengono campionati attraverso il metodo No-U-Turn Sampler, che rappresenta un miglioramento rispetto all'Hamiltonian Monte Carlo [1].

```
# sequential sampling, 2 chain in 1 job
trace = pm.sample(1500, cores=-1, step = pm.NUTS())
trace = trace[250:]
trace = pm.trace_to_dataframe(trace)
```

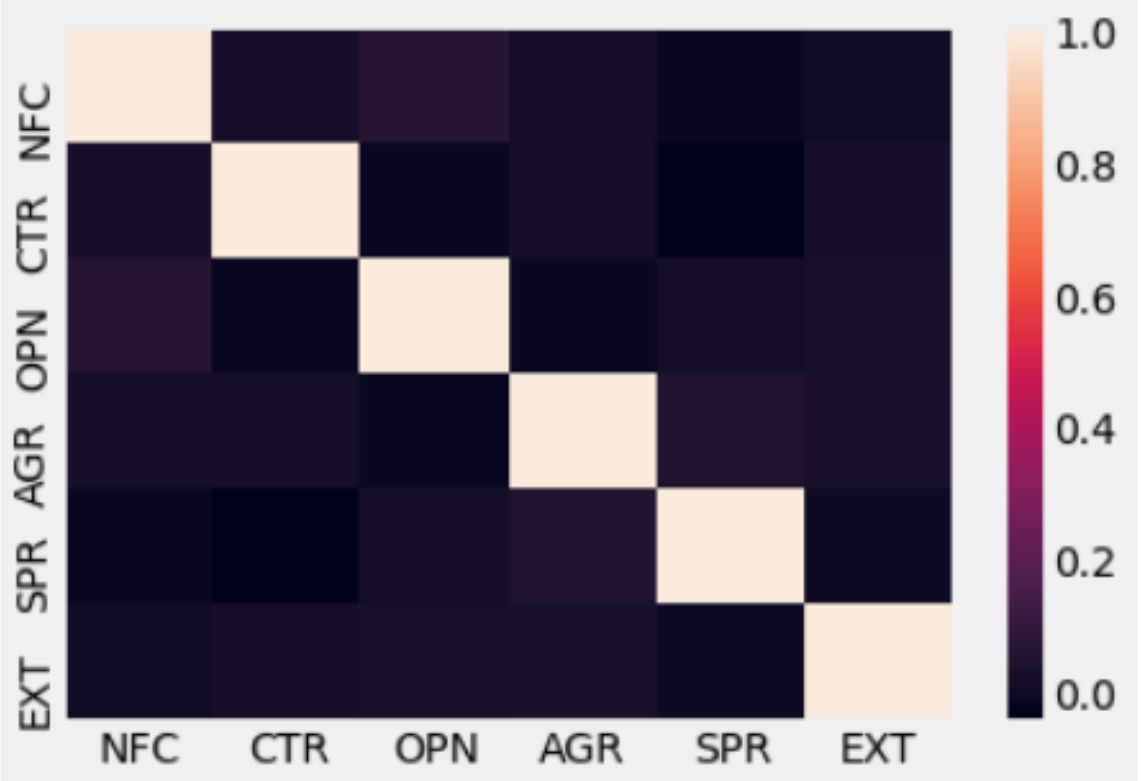
Risultati (un solo topic)

Distribuzione dei campioni per difficoltà di ogni pagina (Question) e abilità di ogni soggetto (Person).



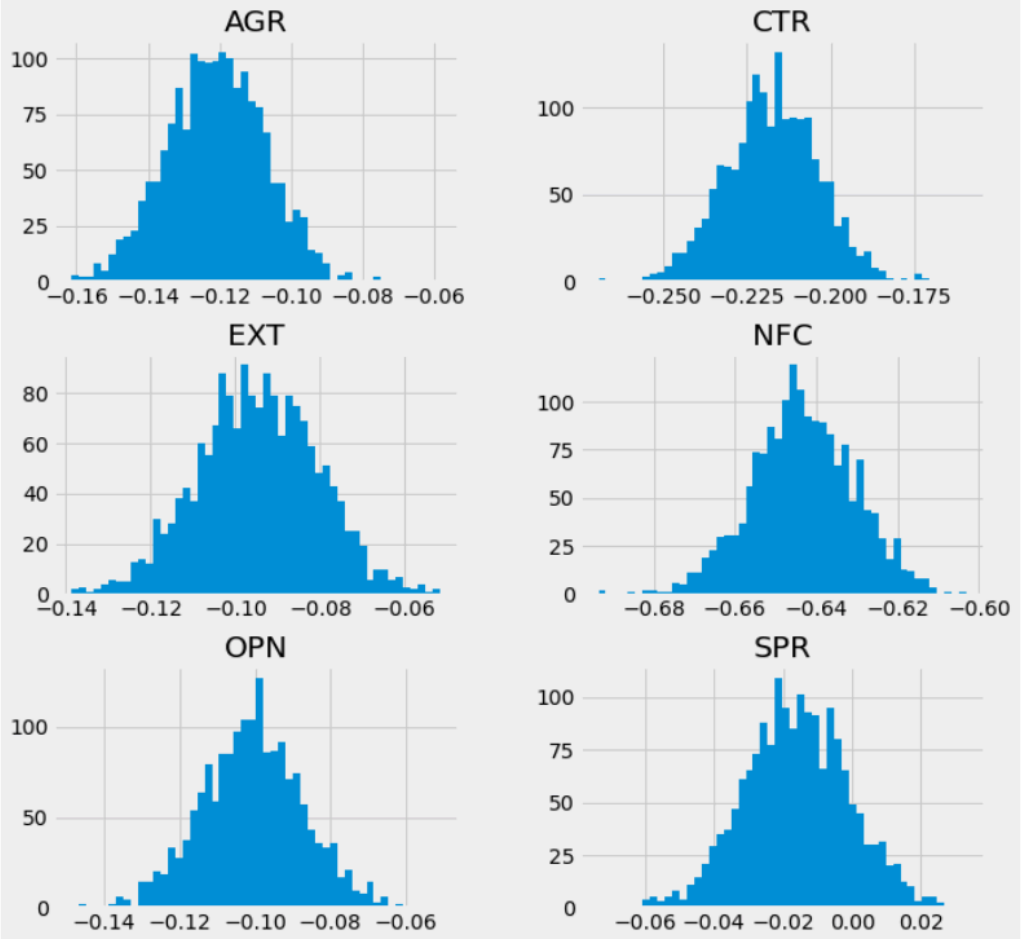
Risultati (fra topics)

La correlazione fra l'abilità (interesse) dei soggetti attraverso diversi topic appare come molto bassa.



Risultati (fra topics)

La distribuzione dell'abilità nei diversi topic assume una distribuzione simil-normale.



Possibili conclusioni

- Abbiamo stimato l'interesse di ogni clienti per ciascun topic editoriale del sito web.
- La stima tiene conto della diversità fra i clienti e della diversa probabilità di visitare ciascuna pagina.

Domande aperte

- I priors scelti hanno senso? Come scegliere i prior in questa situazione?
- Il procedimento di stima e campionamento ha senso? Può essere migliorato (o velocizzato?)
- Originaramente, la formula di Rasch è espressa per dati dicotomici, ma i miei dati non lo sono. Commenti?
- Si può fare qualche commento sulla bontà della stima osservando i grafici?

Grazie della vostra attenzione

Contatti:
alberto.arletti@studenti.unipd.it

References

Hoffman, M. D., Gelman, A., “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo,” *arXiv*, 1111, 4246, 2011