

Report sul funzionamento di *recalculate_user* della libreria Implicit

Andrea Ricci, Lorenzo Palazzetti

1 Introduzione

Il metodo *recommend* di implicit mette a disposizione la flag *recalculate_user*. Secondo quanto riportato nella documentazione ufficiale della libreria, *recalculate_user* non fa affidamento sulle informazioni inerenti l'utente memorizzate nel modello per restituire una raccomandazione, ma la effettua sfruttando esclusivamente i dati contenuti all'interno della matrice sparsa passata come parametro.

Facendo riferimento ad un thread Github, lo sviluppatore della libreria consiglia di utilizzare questo flag in presenza di user o item nuovi; nello specifico user o item non presenti nel dataset al momento dell'allenamento del modello.

Si osserva che l'algoritmo di allenamento del modello è *Alternating Least Squares* nella sua versione *user-based* in quanto l'implementazione dell'algoritmo *Bayesian Personalized Ranking* della libreria implicit non prevede il flag in questione.

2 Test

Per verificare il funzionamento del metodo sono stati effettuati dei test esaustivi facendo variare il valore di *recalculate_user* e il contenuto del dataset. Per modificare quest'ultimo viene creato un dataframe contenente le nuove transazioni, il quale viene concatenato al dataset utilizzato in precedenza per l'allenamento del modello. Di seguito viene riportato un esempio di concatenazione, dove viene aggiunta una sola transazione.

```
data_append = pd.DataFrame({"UserId": ["1109084"], "ItemId": ["922965912"],
                             "Name": ["PR050+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
                             "Quantity": ["5"]})
frames = [data, data_append]
data = pd.concat(frames, ignore_index = True)
```

Viene poi passato questo nuovo dataframe come parametro della funzione *recommend* di Implicit.

2.1 Test con oggetti esistenti

In assenza di oggetti non presenti nel modello non si manifestano problematiche. Si hanno due possibili casi:

- **Utente presente nel modello:** il metodo restituisce raccomandazioni coerenti. Infatti se il flag *recalculate_user* assume valore *True* i suggerimenti tengono conto delle informazioni contenute all'interno del carrello corrente [Es.1], mentre se il flag assume valore *False* il metodo restituisce un risultato che non tiene conto dei suddetti acquisti [Es.2].
- **Utente non presente nel modello:** all'interno di questo scenario si hanno due casi distinti. Se il flag assume valore *True* le raccomandazioni tengono conto delle informazioni contenute all'interno del carrello corrente, restituendo quindi suggerimenti coerenti basati unicamente sugli acquisti eseguiti al momento corrente [Es.3].

In questa circostanza risulta interessante notare che l'ID del nuovo utente potrebbe alterare la corrispondenza fra gli ID reali ed i codici *Category* assegnati all'interno del modello. Nel caso in cui l'ID dell'utente sia successivo a tutti gli altri presenti nel modello gli verrà assegnato il primo codice libero disponibile, ovvero l'ultimo utilizzato + 1. Ad esempio se si hanno 10000 utenti nel modello e ne viene aggiunto uno nuovo con ID avente posizione successiva rispetto a tutti gli altri secondo l'ordine lessicografico, gli verrà assegnato come codice il valore 10001. Al contrario, se l'ID utente occupa una posizione intermedia, si assiste ad uno shift di una posizione di tutti i codici che lo seguono. Impiegando un semplice esempio, se all'utente *Giovanni* era assegnato il code 1000, potrebbe accadere che al momento dell'erogazione di una raccomandazione gli venga assegnato il codice 1001, andando a perdere l'associazione originaria tra l'identità di una persona ed il suo codice. Si reputa che questo shifting non sia però rilevante al momento della raccomandazione, in quanto l'unica informazione interessante contenuta nel modello è la corrispondenza tra codice dell'utente e acquisti effettuati, relazione che rimane tuttavia integra.

Se il flag assume valore *False* i risultati sono casuali, in quanto non vi sono dati da considerare per l'erogazione della raccomandazione [Es.4].

2.2 Test con oggetti non esistenti

In presenza di oggetti non esistenti nel modello si incorre in problematiche significative. Si hanno due possibili casi:

- **ID dell'oggetto successivo a tutti gli altri:** viene assegnato all'item il codice *Category* successivo al più grande attualmente esistente nel modello, si incorre in un errore in esecuzione del tipo *out of bounds index* [Es.5].
- **ID dell'oggetto in posizione intermedia:** l'oggetto provoca uno shift di una posizione dei codici degli elementi che lo seguono. Questa è la situazione peggiore in quanto il sistema raccomanda un oggetto basandosi sui codici in suo possesso, che potrebbero però essere differenti da quelli attualmente in uso [Es.6].

3 Proposte di soluzione

Una delle prime difficoltà da affrontare è quella relativa all'erogazione di raccomandazioni per utenti non presenti all'interno del dataset di partenza. Questa situazione si può verificare ad esempio in corrispondenza dell'emissione di una nuova fidelity card.

La problematica si presenta nel momento in cui il valore del flag *recalculate_user* assuma il valore *False*, infatti così facendo il sistema non possiede alcun dato sugli acquisti dell'utente in questione e restituisce raccomandazioni casuali. Una possibile strategia potrebbe essere quella di mantenere un *set* contenente gli ID degli utenti utilizzati per l'allenamento del modello con cui poter verificare la presenza o meno di un certo ID, in modo da forzare il valore del flag a *True* quando il cliente non fa parte del dataset di partenza. Dal punto di vista computazionale questo approccio risulta essere praticabile, anche in presenza di dataset ampi, in quanto la ricerca di un elemento all'interno della struttura dati *set* viene esautorata con un costo asintotico di $O(\log n)$. Un'idea alternativa sarebbe quella di lasciare permanentemente il valore del flag a *True*, dal momento che ciò non provoca rallentamenti significativi dell'esecuzione ed aiuta a fornire raccomandazioni più precise, possedendo un maggior numero di dati da cui attingere.

Per quanto riguarda gli oggetti la situazione è più delicata in quanto la presenza di un elemento nuovo inficia il corretto funzionamento del sistema. Questa può provocare infatti un errore in esecuzione o addirittura la produzione di raccomandazioni errate e di conseguenza non adatte al cliente. Si ritiene meno rilevante la risoluzione di questo problema in quanto si presuppone che ogni oggetto posto in vendita da un esercizio commerciale sia presente nel suo database. Per poter intercettare questa eventuale situazione, magari dovuta a guasti o malfunzionamenti, si potrebbe applicare lo stesso meccanismo basato sul mantenimento di un *set* di ID utilizzato per gli utenti, in modo tale da sollevare un'eccezione al verificarsi del suddetto evento.

4 Risultati

4.1 Esempio 1

Oggetti esistenti, utente esistente e *recalculate_user* = True

Nell'esempio seguente sono state aggiunte tre transazioni associate ad un utente esistente che acquista molti oggetti appartenenti alla categoria SPORT. Queste transazioni fanno sì che vengano raccomandati elementi facenti parte della stessa categoria (l'utente non aveva mai acquistato oggetti sportivi in precedenza).

```
data_append = pd.DataFrame({"UserId": ["908943"], "ItemId": ["922965912"],
    "Name": ["PRO50+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["500"]})
data_append2 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["903205437"],
    "Name": ["COMPETITION ARANCIA ROSSA 500G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["600"]})
data_append3 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["921222028"],
    "Name": ["CAPELLI FISIOSH RISTR 200ML NF"], "CatCode": ["6AA1X"], "CatName": ["SHAMPOO (ALTRI)"],
    "Quantity": ["200"]})
```

	products Id	products code	products name	Cat Code	Cat name	score
0	971693712	41200	NATURAL MIX BARRETTA 30G	4AA2F16	SPORT	2.599904
1	902169248	2476	CARBO+ BARR ENERG FRUT BOS 40G	4AA2F16	SPORT	2.234949
2	935010052	29712	TOTAL ENERGY STRONG LE GEL40ML	4AA2F16	SPORT	2.022611
3	923130951	15096	MALTO+ GEL ARANCIO 50ML	4AX9	ALIMENTI (ALTRI)	2.000605
4	935487963	30495	NATURAL BOOST BARR FICHI-NOCI	4AA2F16	SPORT	1.979007
5	903527036	4100	X-PLOSIV GEL ARANCIA 30ML	4AA2F16	SPORT	1.977122
6	935010049	29711	TOTAL ENERGY CARBO GEL AGRUMIX	4AA2F16	SPORT	1.902693
7	923876458	16299	LIQUID CARBO+ FLASH 30G ARA	4AA2F16	SPORT	1.888501
8	923131128	15105	X-PLOSIV GEL LEMON-COLA 30ML	4AA2F16	SPORT	1.862212
9	934446283	28685	ENERGYBAR APRICOT BARRETTA 35G	4AX9	ALIMENTI (ALTRI)	1.838746

4.2 Esempio 2

Oggetti esistenti, utente esistente e *recalculate_user* = False

Nell'esempio seguente sono state aggiunte tre transazioni associate ad un utente esistente in cui quest'ultimo acquista molti oggetti appartenenti alla categoria SPORT, ma essendo la flag settata a False vengono raccomandati oggetti unicamente basandosi sugli acquisti passati dell'utente.

```
data_append = pd.DataFrame({"UserId": ["908943"], "ItemId": ["922965912"],
    "Name": ["PRO50+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["500"]})
data_append2 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["903205437"],
    "Name": ["COMPETITION ARANCIA ROSSA 500G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["600"]})
data_append3 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["921222028"],
    "Name": ["CAPELLI FISIOSH RISTR 200ML NF"], "CatCode": ["6AA1X"], "CatName": ["SHAMPOO (ALTRI)"],
    "Quantity": ["200"]})
```

	products Id	products code	products name	Cat Code	Cat name	score
0	908960305	9405	SAUGELLA DERMOLIQ 500ML	6AD5A	DETERGENTI PER IGIENE INTIMA	1.589828
1	926673892	21220	LFP ENTEROTAB 24CPS	4AA2F35	FERMENTI LATTICI	0.912824
2	923208906	15161	THEAL0Z DUO SOL OCULARE 10ML	7BB3A	GOCCE E GEL OCULARI	0.875804
3	923139505	15125	LIPOSCUDIL PLUS 30CPS	4AA2F14	CONTROLLO COLESTEROLO	0.859742
4	931660981	25865	ES0XX ONE 20STICK 10ML	7BB1	PRODOTTI PER APPAR. GASTROINT.	0.801459
5	933904981	27966	DETERGENTE INTIMO FISIOL 400ML	6AD5A	DETERGENTI PER IGIENE INTIMA	0.784175
6	905429318	6375	RDA VIT PLUS 20BUST 160G	4AA2F15	MINERALI/VIT./AMINOACI./PROT.	0.772282
7	927091227	21846	GRINTUSS ADULTI POLISERIN 180G	7BB2C	PRODOTTI PER LA GOLA	0.737480
8	935538494	30663	DIABEC 20CPS	4AA2F99	POLIVALENTI	0.731525
9	931590576	25684	CARTI JOINT FORTE 20CPR	4AA2F09	APPARATO MUSCOLO SCHELETRICO	0.696852

4.3 Esempio 3

Oggetti esistenti, utente non esistente e *recalculate_user* = True

Nell'esempio seguente sono state aggiunte tre transazioni associate ad un utente non esistente in cui quest'ultimo acquista molti oggetti appartenenti alla categoria SHAMPOO e qualche elemento della categoria SPORT, facendo sì che vengano raccomandati elementi facenti parte della categoria con più acquisti.

```
data_append = pd.DataFrame({"UserId": ["842024"], "ItemId": ["922965912"],
    "Name": ["PRO50+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["5"]})
data_append2 = pd.DataFrame({"UserId": ["842024"], "ItemId": ["903205437"],
    "Name": ["COMPETITION ARANCIA ROSSA 500G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["6"]})
data_append3 = pd.DataFrame({"UserId": ["842024"], "ItemId": ["921222028"],
    "Name": ["CAPELLI FISIOSH RISTR 200ML NF"], "CatCode": ["6AA1X"], "CatName": ["SHAMPOO (ALTRI)"],
    "Quantity": ["200"]})
```

	products Id	prod. code	products name	Cat Code	Cat name	score
0	900283615	425	BAGNO DOCCIA SH POMP 300ML	6AD1	BAGNO-DOCCIA	1.944856
1	901338842	1669	BAGNO DOCCIA NUTR LATTE 400ML	6AD1	BAGNO-DOCCIA	1.812076
2	920081357	12144	BAGNO INTIMO FISIOLOGICO LIC	6AD5X	IGIENE INTIMA (ALTRI PRODOTTI PER)	1.533490
3	921222016	12874	CAPELLI FISIOSH DELICATO COLL	6AA1X	SHAMPOO (ALTRI)	1.475125
4	920885427	12583	BAGNO FISIODET N/SAP 400ML	6AD1	BAGNO-DOCCIA	1.207412
5	971043791	38267	LFP DETERGENTE INT LENIT 250ML	6AD5A	DETERGENTI PER IGIENE INTIMA	1.181753
6	909741581	9920	CORPO CREMA MANI MIELE 75ML	6AE4A	IDRATANTI/NUTRIENTI PER LE MANI	1.014643
7	920081332	12143	BAGNO INTIMO ATTIVO 250ML	6AD5X	IGIENE INTIMA (ALTRI PRODOTTI PER)	0.958696
8	927305565	22838	LFP DEOSPRAY ANTIMICROBICO	6AD2	DEODORANTI CORPO/ANTITRASPIRANTI	0.953472
9	903138446	3860	BAGNO DELICATO CALENDULA 400ML	6AC5A	DETERGENTI/STRUCCANTI/TONICI VISO	0.947099

4.4 Esempio 4

Oggetti esistenti, utente non esistente e *recalculate_user* = False

Nell'esempio seguente sono state aggiunte tre transazioni inerenti un utente non esistente, ma essendo il flag settato a false non vengono prese in considerazione. I suggerimenti di acquisto erogati non forniscono quindi un'informazione interessante, dato avvalorato dai bassi valori assegnati agli score.

```
data_append = pd.DataFrame({"UserId": ["842024"], "ItemId": ["922965912"],
    "Name": ["PRO50+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["5"]})
data_append2 = pd.DataFrame({"UserId": ["842024"], "ItemId": ["903205437"],
    "Name": ["COMPETITION ARANCIA ROSSA 500G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
    "Quantity": ["6"]})
data_append3 = pd.DataFrame({"UserId": ["842024"], "ItemId": ["921222028"],
    "Name": ["CAPELLI FISIOSH RISTR 200ML NF"], "CatCode": ["6AA1X"], "CatName": ["SHAMPOO (ALTRI)"],
    "Quantity": ["200"]})
```

	products Id	prod. code	products name	Cat Code	Cat name	score
0	933151209	27234	BETOTAL PLUS 40CPR PROMO	4AA2F15	MINERALI/VITAMINE/AMINOACIDI/PROT.	0.017568
1	925512636	18658	NEOBIANACID 45CPR MASTICABILI	7BB1	PRODOTTI PER APPARATO GASTROINTES.	0.015431
2	902557824	3159	ENTEROLACTIS PLUS 20CPS	4AA2F35	FERMENTI LATTICI	0.011458
3	926025913	19429	MERITENE CIOCCOLATO 270G	4AA2B99	ALIMENTI DESTINATI A FINI SPECIALI	0.010233
4	925038996	18082	ENTEROLACTIS 12FL 10ML	4AA2F35	FERMENTI LATTICI	0.009532
5	932183193	26487	POLICOL 1000 30CPR	4AA2F99	POLIVALENTI	0.009256
6	935662609	31149	SUPRADYN RIC DEGLUT 35CPR	4AA2F15	MINERALI/VITAMINE/AMINOACIDI/PROT.	0.009175
7	912039385	10844	ENTERELLE 12CPS	4AA2F35	FERMENTI LATTICI	0.008574
8	902085986	2417	MAGNESIO SUPREMO POLVERE 150G	4AA2F15	MINERALI/VITAMINE/AMINOACIDI/PROT.	0.008419
9	972295671	42752	CONNETTIVINABIO PLUS CREMA 25G	7BB7A	PRODOTTI PER LA CUTE	0.008244

4.5 Esempio 5

Oggetti non esistenti e *recalculate_user* = True (ID in ultima posizione)

Nell'esempio seguente viene assegnato all'oggetto non esistente un codice Category successivo rispetto a tutti gli altri, fattore che porta ad un errore in esecuzione.

```
data_append = pd.DataFrame({"UserId": ["908943"], "ItemId": ["922965912"],
                             "Name": ["PRO50+ CIOCCOLATO BIANCO 50G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
                             "Quantity": ["5"]})
data_append2 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["903205437"],
                              "Name": ["COMPETITION ARANCIA ROSSA 500G"], "CatCode": ["4AA2F16"], "CatName": ["SPORT"],
                              "Quantity": ["6"]})
data_append3 = pd.DataFrame({"UserId": ["908943"], "ItemId": ["999999999"],
                              "Name": ["OGGETTO NON ESISTENTE"], "CatCode": ["00000"], "CatName": ["NOTEX"],
                              "Quantity": ["1"]})
```

Traceback (most recent call last):

```
File "load_als.py", line 281, in <module>
    recommended = model.recommend(user_id, sparse_user_item, N=20, recalculate_user=True)
File "implicit/recommender_base.pyx", line 159, in implicit.recommender_base.MatrixFactorizationBase.recommend
File "implicit/recommender_base.pyx", line 304, in implicit.recommender_base.MatrixFactorizationBase._user_factor
File "C:\Users\Ricci\AppData\Roaming\Python\Python37\site-packages\implicit\als.py", line 216, in recalculate_user
    self.regularization, self.factors)
File "C:\Users\Ricci\AppData\Roaming\Python\Python37\site-packages\implicit\als.py", line 346, in user_factor
    A, b = user_linear_equation(Y, YtY, Cui, u, regularization, n_factors)
File "C:\Users\Ricci\AppData\Roaming\Python\Python37\site-packages\implicit\als.py", line 333, in user_linear_equation
    factor = Y[i]
```

IndexError: index 50822 is out of bounds for axis 0 with size 50822

4.6 Esempio 6

Oggetti non esistenti e `recalculate_user = True` (ID intermedio)

Nell'esempio seguente il codice Category assegnato dell'oggetto non esistente non è quello in ultima posizione, il che porta ad uno scorrimento dei codici degli altri oggetti.

```
data_append = pd.DataFrame({"UserId": ["908943"], "ItemId": ["904553095"],  
    "Name": ["OGGETTO NON ESISTENTE"], "CatCode": ["NOTEX"], "CatName": ["NOTEX"],  
    "Quantity": ["1"]})
```

Le tabelle seguenti rappresentano le raccomandazioni erogate per lo stesso utente prima e dopo l'inserimento del nuovo item. Com'è possibile notare l'oggetto corrispondente al codice **9405** è **SAUGELLA DERMOLIQ 500ML** all'interno del modello, ma diventa **SAPONE ZOLFO PH5 100G** in seguito all'immissione. Questo porta a raccomandazioni fuorvianti.

	products Id	prod. code	products name	Cat Code	Cat name	score
0	908960305	9405	SAUGELLA DERMOLIQ 500ML	6AD5A	DETERGENTI PER IGIENE INTIMA	1.387963
1	905429318	6375	RDA VIT PLUS 20BUST 160G	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	1.295700
2	904258720	5016	GOLACALM SPRAY 20ML	4AA2F23	GOLA	1.204920
3	904570886	5410	LINEAMED CIOCC/MAND BAR 50G	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	0.997155
4	912166206	10905	LINEAMED FRAG/YOGURT BAR 50G	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	0.941709
5	908009917	8549	ACQUA TABIANO PERGOLI 1LT	5AB	ACQUE MINERALI PER USO TOPICO	0.878064
6	922326917	13857	LINEAMED BARR COCCO/CIOCC 60G	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	0.877071
7	922326956	13860	LINEAMED BARR CAFFE/CIOCC 60G	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	0.871614
8	935661239	31140	LDF HIDRASAL MAG+POT 20BUST	4AA2F15	MINER./VITAMINE/AMINOACIDI/PROT.	0.869131
9	909272458	9655	VITAMINDERMINA POLV 100G	7AF6C01	TALCHI PER BAMBINO	0.758370

	products Id	prod. code	products name	Cat Code	Cat name	score
0	905429142	6375	FLOLIGO GEM 16 GOCCE 30ML	4AA2F99	POLIVALENTI	1.377715
1	908958073	9405	SAPONE ZOLFO PH5 100G	6AD1	BAGNO-DOCCIA	1.352825
2	904258720	5016	GOLACALM SPRAY 20ML	4AA2F23	GOLA	1.133371
3	904570658	5410	ALVITA OLIO MANDORLE DOLCI 50ML	3AC3A99	FORMULAZ. FITOT. PRONT	0.950619
4	938976343	33450	KERAMINE H FASC RO PROMO	6AA2A	LOZIONI PER CAPELLI	0.938464
5	912164884	10905	RESPECTISSIME LINER NERO	6AC2B	MATITE/EYE LINER	0.898313
6	935661227	31140	EAU DE JEUNESSE TRATT RASS P M	6AC5I03	ANTIRUGHE PER LA PELLE DEL VISO	0.866055
7	908009448	8549	GERME GRANO CONC OLEOSO 100ML	4AA2F08	APPARATO DIGERENTE	0.856806
8	922326881	13857	ACTEEN HYDRAMAT CR SEBONORMAL	6AC5I02	SEBOREG. PER LA PELLE DEL VISO	0.839443
9	922326931	13860	DEFENCE B-LUCENT CR VISO SPF15	6AC5I05	SCHIARENTI PER LA PELLE DEL VISO	0.836672