UNIVERSITÀ DI PISA
Dipartimento di Informatica
Corso di laurea triennale di Informatica

STUDIO E CONFRONTO DI METRICHE PER SISTEMI DI RACCOMANDAZIONE SU DATI UNARI

Relatore:
Prof. *Francesco Romani*

Autore:
Gemma Martini

Correlatore:
Prof. *Gianna Maria Del Corso*

ANNO ACCADEMICO 2017–2018

Indice

Un grazie di cuore
a tutti coloro che
mi hanno supportato
in questo periodo
decisivo.

Introduzione

Nella società di oggi, le compagnie dispongono di una grande quantità di dati sugli utenti di molteplici piattaforme online e applicazioni mobile, quali i *feedback* su prodotti e servizi offerti. Processare questi dati mediante algoritmi opportunamente progettati consente di produrre *suggerimenti* per acquisti futuri.

Nel novero dei metodi di analisi dei dati sopra descritti si trovano approcci brutali e fastidiosi, quali proporre prodotti e contenuti uguali a quelli già graditi (o acquistati) o, se non uguali, estremamente simili in natura ed utilizzo. Questo procedimento sfocia nel fenomeno secondo il quale se un utente ha acquistato un determinato articolo gli vengono proposti articoli molto simili, anche se presumibilmente, una volta acquistato un prodotto, difficilmente si sarà interessati a ripetere il medesimo acquisto nel giro di un tempo breve.

Il problema di fornire raccomandazioni ad utenti dei quali si è in grado di tracciare un "profilo" può essere approcciato anche proponendo agli utenti suggerimenti che non siano soltanto funzione degli acquisti precedenti, ma che si basino anche sugli acquisti fatti da utenti *simili*. Questi *sistemi di raccomandazione* si suddividono in tre categorie [3, 15]:

Filtri collaborativi, che si basano sulla creazione di un profilo per ogni utente per poi fornirgli suggerimenti ottenuti dal confronto dell'utente *target* con gli altri utenti del *dataset*. In questo modello non si richiede uno studio dei contenuti, oggetto delle preferenze;

Filtri basati su contenuto i quali si focalizzano sul profilo utente, ma anche su uno studio di caratteristiche intrinseche dei dati, così da poter suggerire ad un utente con determinate preferenze articoli che appartengono alle categorie alle quali è interessato;

Sistemi ibridi in cui l'attenzione è focalizzata sulla combinazione di vari approcci tra cui i due approfonditi in precedenza. In particolare, i sistemi di raccomandazione ibridi di tipo contenuto/collaborativo mostrano problemi legati alla necessità di avere molte valutazioni sugli oggetti, sebbene siano i sistemi ibridi più largamente implementati.

Uno dei dataset più utilizzati per effettuare esperimenti sui sistemi di raccomandazione è quello di MovieLens [6], che raccoglie i rating assegnati dagli utenti ad alcuni film. Di questo dataset sono state prodotte diverse versioni, che presentano un

numero diverso di valutazioni, da 100 000 a 10 000 000. Altri dataset sono descritti in Appendice B.

Dalle caratteristiche di questo dataset emerge una delle difficoltà con cui si scontrano i sistemi di raccomandazione nel mondo reale, ovvero la quantità di dati che devono gestire. Si pone dunque un *problema* fondamentale: utilizzare il *minor numero di risorse* (in termini di memoria utilizzata, tempo di calcolo e conseguente dispendio energetico) per produrre risultati tutto sommato soddisfacenti. È interessante notare che, in alcune situazioni estreme, progettare algoritmi efficienti non è abbastanza per ottenere tempi di calcolo ragionevoli. In alcuni casi, infatti, la quantità di dati è talmente alta (si pensi a Google o Facebook) che anche il più efficiente degli algoritmi di raccomandazione impiegherebbe giorni per produrre risultati. In un contesto come questo si sono sviluppati algoritmi che risolvono il problema di *aggiornare i suggerimenti* man mano che arrivano nuovi dati, senza sprecare risorse ricalcolando tutto da capo. Questo approccio non sarà trattato in questa tesi.

Data l'enorme vastità di contenuti offerti dai sistemi trattati, inevitabilmente, per ogni utente, la *percentuale* di elementi sui quali si hanno informazioni è *minima*. Inoltre, non è chiaro se l'assenza di valutazione sottintenda un giudizio negativo: ad esempio, è probabile che un fan di film rosa non valuti film horror perché non ritiene che gli possano piacere; d'altra parte probabilmente non ha nemmeno valutato tutte le migliaia di film rosa esistenti, anche solo per mancanza di tempo.

Per gestire questo scenario, in letteratura sono state seguite sia la strada della non-valutazione "neutra" che quella della non-valutazione "negativa": infatti, la maggior parte dei lavori assumono implicitamente che le valutazioni presenti siano un campione casuale di tutte quelle possibili, mentre altri (come [11, 12, 13]) assumono che sia più probabile che una valutazione mancante sia indicativa di un'opinione dell'utente.

La letteratura ha approfondito algoritmi e metriche di valutazione su dati che rappresentano preferenze in un certo range $[0, k]$, come per esempio in [2, 8, 9], ma è ancora relativamente inesplorato il problema di ricavare informazioni da dati "unari".

Questa tesi si pone come obiettivo quello di offrire un confronto sulle misure, da intendersi come *preparatorio* per una trattazione più ampia sui sistemi di raccomandazione su dati unari, alla stregua di quanto prodotto in [7] per dati non unari. Si ritiene infatti necessario fornire standard ai quali attenersi per valutare nuovi algoritmi, poichè l'assenza completa di metri di paragone universali rallenta ed inficia non poco la ricerca nell'ambito dei sistemi di raccomandazione.

L'esempio che meglio rappresenta il problema "unario", ovvero il caso approfondito in questa tesi, è il noto *social network* Facebook, in cui un utente può visualizzare un contenuto (cosiddetto *post*) e assegnargli un "mi piace" (*like*). In questo modello, i dati a disposizione sono estremamente limitati: infatti non è possibile sapere se l'assenza di un *like* sia dovuta ad una mancata visualizzazione del contenuto oppure ad uno scarso gradimento dello stesso.

In questa tesi si è dunque scelto di

- trattare dati *unari*, ossia matrici in cui la cifra 1 rappresenta che l'articolo è stato gradito (o acquistato), mentre la cifra 0 indica che l'articolo non è stato votato o

perché tale utente non ne era interessato o perché non ne conosceva l'esistenza

e di

- limitarsi allo studio di algoritmi basati sulla *fattorizzazione di matrici*,

con l'obiettivo di adattare le metriche esistenti per valutazioni esplicite al caso di dati unari, in modo da aprire la strada per lo sviluppo di algoritmi *ad hoc* per il problema unario.

Prima di entrare nel vivo nella trattazione si ritiene cruciale chiarire l'uso dei seguenti termini:

- **Sparsità** (o **densità**) 1 intende che la matrice è "piena";
- **Sparsità** (o **densità**) 0 intende che la matrice è "vuota".

Più nel dettaglio, si offre un organigramma della trattazione:

Capitolo 1 — Metriche: In questa prima parte si espongono le misure usate più largamente in ambito accademico per valutare la bontà di un recommendation algorithm. Dopo una prima esposizione, tali misure vengono adattate al problema unario e, laddove possibile, al problema unario risolto mediante la creazione di matrici approssimate anch'esse di tipo unario;

Capitolo 2 — Cenni sui metodi risolutivi: È proprio in questo capitolo che vengono approfonditi tra gli altri due metodi per produrre raccomandazioni: SVD e NMF. La presentazione degli algoritmi è accompagnata da dimostrazioni formali circa la correttezza del procedimento;

Capitolo 3 — Sperimentazioni sulle misure: In questo capitolo si forniscono i risultati ottenuti dai confronti operativi tra le varie misure e si espongono i metodi utilizzati per sperimentare. Questo è il vero e proprio fulcro della tesi, poichè illustra come scegliere tra classi di "equivalenza tra metriche", che si adattino agli scopi specifici di un determinato sistema di raccomandazione;

Capitolo 4 — Conclusioni: È in questa parte che si tirano le fila del lavoro svolto, fornendo spunti per lavori futuri.

Il resto del capitolo corrente è dedicato a spiegare l'intuizione che sta dietro agli algoritmi basati su fattorizzazione di matrici.

Latent semantic indexing

Il lettore può notare che anche nella vita del più ordinario degli esseri umani è presente la nozione astratta di *sistema di raccomandazione*, tanto che ne usa uno quando, in presenza di altri individui, si appresta a suggerire un'attività da fare durante l'incontro.

Si noti, ad esempio, questo scenario: un piccolo gruppo di persone si ritrova per una serata al cinema e deve trovare un film tollerabile, se non gradito a tutti. Si conoscono i gusti delle persone (per esempio, qualcuno adora i film d'amore, ma non sopporta gli horror) e si conoscono i generi dei film; da questo si riesce quindi a dedurre quanto a una persona possa interessare un film e, di conseguenza, fare una proposta che provi ad accontentare tutti.

Ovviamente, questo meccanismo è difficile da applicare nel caso in cui vi siano milioni di film e di utenti. Partendo però dall'assunzione che gli utenti diano un giudizio su un film in base al proprio gradimento delle *features* di ogni film, si possono costruire sistemi di raccomandazione che sfruttano questo principio su dati reali [9].

Questo processo di individuazione delle features rappresentative delle caratteristiche dei dati è noto in letteratura come *latent semantic indexing* e l'insieme delle features associate a un certo utente o contenuto è noto come *latent factor*.

Più nel dettaglio, fissato un numero di features k, si cerca un vettore w_l di taglia k per ogni utente u_l e un altro vettore h_h, sempre di taglia k, per ogni contenuto i_h.

La costruzione di questi vettori è funzionale all'approssimazione della matrice che rappresenta il dataset (detta matrice A, meglio definita in Preliminari) espressa come prodotto di feature nascoste. In termini più formali, l'assunzione del latent semantic indexing è quindi che il voto dell'utente u_l sul contenuto i_h sia scrivibile come $v_l^T w_h$. Di conseguenza, detta W la matrice che ha per righe i vettori w_l e H la matrice che ha per righe i vettori h_h, la matrice dei giudizi A può essere approssimata da WH^T.

Si osserva che il prodotto WH^T ha rango k e non è detto che la matrice dei giudizi A abbia a sua volta rango k. I metodi di fattorizzazione di matrici (approfonditi nel Capitolo 2)si propongono quindi di trovare due matrici W e H che forniscono una buona approssimazione di A secondo una qualche funzione errore: per esempio, se si vuole minimizzare $\|A - WH^T\|_F$*, è noto che si riesce a trovare una soluzione con la *singolar value decomposition*, descritta nella Sezione 2.1.

Sono noti in letteratura (cfr. [1, 10]) molti metodi per ottenere decomposizioni di rango k approssimate, modificando la funzione da minimizzare o anche imponendo vincoli diversi sulle matrici W e H. Una delle varianti più note di questo schema è quello che impone che tutti i valori nelle matrici W e H siano non-negativi, cioè secondo la seguente intuizione: un film può non possedere una certa feature, ma non può "possederne una quantità negativa". Questa idea ha portato a numerosi algoritmi, il più semplice dei quali minimizza $\|A - WH^T\|_F$ ed è noto come NMF, descritto nella Sezione 2.2.

In generale, questi metodi si scontrano con numerose difficoltà nel caso di matrici unarie. Infatti, in questo particolare caso, tutti i valori noti sono 1. Esiste quindi un'approssimazione di rango 1 che fornisce errore nullo su tutti i dati, ovvero quella che assegna un 1 ad ogni possibile valutazione. Per quanto questa approssimazione sia perfetta dal punto di vista della funzione di errore $\|A - WH^T\|_F$, poichè calcolata tenendo conto solo delle valutazioni conosciute, è evidente come abbia poco valore predittivo.

*dove $\|\cdot\|_F$ si chiama *norma di Frobenius* ed è definita come la radice quadrata della somma dei quadrati degli elementi della matrice

Si rende quindi necessario introdurre termini di normalizzazione nelle funzioni errore, proprio per evitare questi problemi.

Ulteriori dettagli su alcuni algoritmi risolutivi e possibili loro adattamenti si possono trovare nel Capitolo 2.

Preliminari

Come accennato nell'introduzione, i recommendation system hanno come dato di input una matrice che ha per righe gli utenti della piattaforma e per colonne i contenuti. Ogni elemento della matrice rappresenta se e quanto il contenuto in oggetto sia gradito dal tale utente.

Lo scopo di un algoritmo di raccomandazione è quello di diminuire la sparsità della matrice, ottenendo come output una matrice dello stesso tipo di quella di input, ma con un numero superiore di cifre diverse da zero.

Per permettere al lettore di seguire la trattazione in modo fluido si è scelto di soffermarsi sulla notazione nei pochi pargrafi seguenti.

Nelle piattaforme in oggetto si parla di *utenti* $\mathcal{U} = \{u_1, u_2, \ldots, u_m\} \in \mathbb{R}^n$ ed *elementi* (dall'inglese *items*) $\mathcal{I} = \{i_1, i_2, \ldots, i_n\} \in \mathbb{R}^m$.

La matrice dei dati per il nostro problema è la *matrice di utilità* o matrice di ranking $A \in M(m, n, \mathcal{V})$, ossia appartenente allo spazio delle matrici di m righe ed n colonne a valori nell'insieme $\mathcal{V}$. Tale insieme $\mathcal{V}$ può essere $\{1, 2, 3, 4, 5\}$, come nel caso di Netflix fino a poco tempo fa, oppure $\{-1, 0, 1\}$ (dataset binari), che è il tipo di feedback dei quali dispone Netflix al momento attuale, oppure, nel caso del problema unario (di cui si è accennato nell'introduzione), i dati hanno valori appartenenti all'insieme $\mathcal{V} = \{0, 1\}$.

Dove non altrimenti indicato, con l si rappresenta l'indice di un utente e con h si rappresenta l'indice di un elemento. Pertanto, per ogni coppia $(u_l, i_h) \in \mathcal{U} \times \mathcal{I}$ la matrice A è fatta come segue

$$a_{l,h} = \begin{cases} 1 & \text{se l'utente } u_l \text{ ha gradito il contenuto } i_h \\ 0 & \text{altrimenti} \end{cases}$$

Denotiamo $\Omega \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ tale che $\Omega = \{(l, h) \mid a_{l,h} = 1\}$ in cui ogni coppia (l, h) indica che l'utente u_l ha gradito l'elemento i_h. In modo speculare, sia $\overline{\Omega} \subseteq \{1, \ldots, m\} \times \{1, \ldots, n\}$ tale che $\overline{\Omega} = \{(l, h) \mid a_{l,h} = 0\}$.

L'obiettivo di un sistema di raccomandazione è quello di calcolare una matrice $B*$, che rappresenti dei buoni suggerimenti per gli utenti della piattaforma. Questo output può essere di vari tipi, ciascuno con una sua interpretazione:

- con BR indichiamo una matrice composta da numeri reali, in cui a numeri più alti corrisponde un maggiore gradimento dell'elemento. Nel caso particolare in cui

gli elementi della matrice siano nel range $[0, 1]$, una possibile interpretazione alternativa è quella di una matrice contenente *probabilità di gradimento*: ogni posizione rappresenta quanto è probabile che tale elemento piaccia ad una determinata persona;

- con $\mathtt{BD}$ indicheremo una matrice composta di soli 0 e 1, in cui ogni 1 indica che si prevede che all'utente piaccia quel contenuto;
- con $\mathtt{BN}(\mathcal{N})$, con $\mathcal{N} \in \mathbb{N}$ (tipicamente $\mathcal{N} = 10$) indicheremo una matrice composta di soli 0 e 1 e che inoltre soddisfa la seguente proprietà: per ogni utente, esattamente $\mathcal{N}$ caselle che erano 0 in A sono 1 in $\mathtt{BN}(\mathcal{N})$. Questa situazione corrisponde a fornire esattamente $\mathcal{N}$ raccomandazioni a ciascun utente.

Per valutare la "bontà" di un algoritmo di raccomandazione, sono state definite numerose misure. Valutare queste misure può richiedere di esaminare i valori delle caselle delle matrici (è il caso delle *metriche di rating*), l'ordinamento relativo delle raccomandazioni per un certo utente (*metriche di ranking*), oppure quali delle raccomandazioni siano effettivamente corrette (*metriche di classificazione*).

Da queste definizioni si può fare un'osservazione preliminare: non tutte le combinazioni di matrice risultato ($\mathtt{BR}$, $\mathtt{BD}$, $\mathtt{BN}(\mathcal{N})$) e di metrica sono possibili. Questa affermazione è supportata da adeguate prove formali nel Capitolo 1.

Si osservi inoltre che è sempre possibile ottenere una matrice di tipo $\mathtt{BD}$ o $\mathtt{BN}(\mathcal{N})$ a partire da una matrice di tipo $\mathtt{BR}$ con un processo noto come *sogliatura*, di cui saranno forniti maggiori dettagli nella Sezione 2.5.

Si ritiene opportuno anticipare in questa parte prliminare la notazione che accompanerà il lettore per tutta la trattaqzione: indicheremo con $b_{l,h}$ l'elemento di posizione (l, h) in una matrice a coefficienti reali, e con $\tilde{b}_{l,h}$ il corrispondente elemento dopo la sogliatura (dunque, 0 o 1).

Per concludere, si noti che effettuare una sogliatura è necessario quando non si è interessati a mantenere un ordine dei suggerimenti, ma si vuole circoscrivere le proposte ad un insieme di cardinalità fissata, come avviene quando si parla di $\mathtt{BN}(\mathcal{N})$.

Capitolo 1

Metriche

Alla stregua di quanto fatto in [4], in questa tesi si vuole offrire una panoramica di tutte le metriche che valutano la bontà delle raccomandazioni per il problema unario, che è computazionalmente molto diverso da un classico problema di raccomandazione.

In questa trattazione si è scelto di usare come sinonimi i termini *metrica* e *misura*, come avviene in tutta la letteratura del settore, sebbene matematicamente i due termini abbiano definizioni diverse.

All'interno dell'arcinota tecnica di valutazione di un algoritmo *k-fold cross validation* si denota $\mathcal{R}$ il *test set*. In questo capitolo, se non specificato altrimenti, con la matrice A si indica la matrice che contiene i rating appartenenti a $\mathcal{R}$.

Le misure a breve introdotte si dividono in più categorie:

- Metriche di classificazione;
- Metriche di rating;
- Metriche di ranking.

Di seguito si trova una trattazione approfondita di queste tre tipologie, preceduta da una considerazione: le metriche esposte in questa tesi non sono tutte le metriche presenti in letteratura, nè si prefiggono di valutare a tutto tondo la bontà di un algoritmo di raccomandazione. Infatti, gli aspetti da considerare nella valutazione di una raccomandazione sono molti e spaziano tra quantità di casi sui quali allenare gli algoritmi, a quanto sono "originali" nel suggerire oggetti.

Lo scopo delle misure trattate è quello di valutare alcune caratteristiche di una raccomandazione. Si richiede inoltre che le metriche che misurano queste caratteristiche siano *stabili* ed *utilizzabili*. Come puntualizzato in [7], può essere interessante utilizzare una misura quale il **coverage**, ossia percentuale degli elementi presenti nel sistema che viene effettivamente raccomandata. Tuttavia, non è affatto facile da impiegare, poichè un modo per "barare" su questa metrica è quello di inserire raccomandazioni fasulle, diminuendo così l'accuratezza. In conclusione, si è deciso di non trattare tale misura nell'ambito di questa tesi, perchè non esiste ancora una vera e propria definizione di coverage utilizzabile.

1.1 Metriche di classificazione

Questa tipologia di misure pone l'attenzione sulle raccomandazioni fatte, contrapposte a quelle che sarebbero dovute esser fatte. Per come sono definite, **richiedono** intrinsecamente di effettuare un qualche tipo di **sogliatura** sulla matrice dei risultati, anche nel caso di matrice iniziale reale: infatti, queste metriche sono definite su insiemi, ed è pertanto necessario avere ben definito quali elementi appartengono all'insieme dei suggeriti e quali no.

Dato l'insieme degli elementi *rilevanti* $\mathtt{Rel} = \{(l, h) \in \mathcal{R} \mid a_{l,h} \geqslant \sigma_A\}$ (si intende che un articolo i_hè rilevante per un utente u_l se la valutazione data da u_l a i_h supera una certa costante σ_A) e l'insieme degli elementi *predetti* $\mathtt{Pre} = \{(l, h) \in \mathcal{R} \mid \tilde{b}_{l,h} = 1\}$, si definiscono i seguenti insiemi:

veri positivi $\mathtt{tp} = \mathtt{Rel} \cap \mathtt{Pre}$;

falsi positivi $\mathtt{fp} = \mathtt{Pre} \cap (\overline{\mathtt{Rel}} \cap \mathcal{R})$;

veri negativi $\mathtt{tn} = (\overline{\mathtt{Rel}} \cap \mathcal{R}) \setminus \mathtt{fp}$;

falsi negativi $\mathtt{fn} = \mathtt{Rel} \setminus \mathtt{tp}$.

Le misure considerate in questa trattazione sono le seguenti:

- **Precision**: $\mathbf{P} = \frac{|\mathtt{tp}|}{|\mathtt{Pre}|} = \frac{|\mathtt{tp}|}{|\mathtt{tp}|+|\mathtt{fp}|}$;
- **Recall**: $\mathbf{R} = \frac{|\mathtt{tp}|}{|\mathtt{Rel}|} = \frac{|\mathtt{tp}|}{|\mathtt{tp}|+|\mathtt{fn}|}$;
- **F1-score**: $\mathbf{F1} = \frac{2 \cdot \mathbf{P} \cdot \mathbf{R}}{\mathbf{P}+\mathbf{R}}$;
- **Accuracy**: $\mathbf{A} = \frac{|\mathtt{tp}|+|\mathtt{tn}|}{|\mathcal{R}|}$.

1.2 Metriche di rating

Le misure di questo tipo esprimono la "distanza" tra la matrice incognita A e la matrice approssimata B. In questa categoria si annoverano le seguenti:

- **M**ean **A**bsolute **E**rror: $\mathbf{MAE}(A, B) = \frac{1}{|\mathcal{R}|} \sum_{(l,h)\in\mathcal{R}} |a_{l,h} - b_{l,h}|$;
- **R**oot **M**ean **S**quare **E**rror: $\mathbf{RMSE}(A, B) = \sqrt{\frac{1}{|\mathcal{R}|} \sum_{(l,h)\in\mathcal{R}} |a_{l,h} - b_{l,h}|^2}$*.

*La misura RMSE è comunemente usata nell'ambito dell'analisi dei dati, ma è ereditata dalla teoria dei segnali. Un'alternativa è usare una misura diversa, del tipo $\frac{\|A-B\|_F}{|A\|}$, che è una misura di tipo matriciale ed è più naturale in ambiente numerico. Da notare che nel caso di A unaria questi due approcci coincidono.

Osserviamo che, al contrario di tutte le altre metriche descritte in questo capitolo, **MAE** e **RMSE** non hanno valori in $[0, 1]$ ma in $[0, +\infty)$ e che i valori più desiderabili per queste metriche sono quelli vicini a 0, mentre per le altre misure i valori migliori sono quelli vicini a 1.

1.3 Metriche di ranking

Questo tipo di misure valutano quanto un algoritmo predice in modo veritiero l'ordine di raccomandazione degli elementi ad un certo utente. In questa categoria non si tiene conto dei punteggi dei vari elementi e quanto si discostano dal rating effettivo nella matrice da approssimare A, ma conta solo l'ordine degli elementi per ogni utente in base al loro punteggio.

Prima di introdurre formalmente le metriche è necessario dare alcune definizioni preliminari:

$\mathbf{I_r}(l) = \{h \mid (l, h) \in \mathcal{R},\ a_{l,h} \geqslant \sigma_A\}$, insieme degli articoli rilevanti per l'utente u_l (come per le metriche di classificazione, si intende che un articolo è rilevante per un utente se il suo rating su quell'articolo supera una certa costante σ_A);

$\mathbf{U_r} = \{l \mid I_r(l) \neq \emptyset\}$, insieme degli utenti che hanno valutato almeno un elemento;

$\mathbf{R_l} = \{h \mid (l, h) \in \mathcal{R}\}$, insieme degli articoli, ristretto al solo test set;

Φ_l è la permutazione degli elementi di $\mathcal{R}_l$ (riga di $\mathcal{R}$ relativa all'utente u_l), che induce l'ordinamento decrescente $a_{l,\Phi_l(1)} \geqslant a_{l,\Phi_l(2)} \geqslant \cdots \geqslant a_{l,\Phi_l(|\mathcal{R}_l|)}$;

π_l è la permutazione degli elementi di $\mathcal{R}_l$, che induce l'ordinamento decrescente $b_{l,\pi_l(1)} \geqslant b_{l,\pi_l(2)} \geqslant \cdots \geqslant b_{l,\pi_l(|\mathcal{R}_l|)}$;

$\mathbf{pr_N}(l) = |\{h \leqslant N \mid \pi_l(h) \in I_r(l)\}|$ è il numero di elementi veramente rilevanti per l'utente u_l tra gli N suggeriti.

Il collegamento con la precision definita nella Sezione 1.1 è che $pr_N(l)$, se divisa per il numero di suggerimenti, equivale alla *precision*;

$$\mathbf{DCG}(l) = \sum_{j=1}^{|I_r(l)|} \frac{2^{a_{l,\pi_l(j)}} - 1}{\log_2(j+1)} \textit{ discounted cumulative gain};$$

$$\mathbf{IDCG}(l) = \sum_{j=1}^{|I_r(l)|} \frac{2^{a_{l,\Phi_l(j)}} - 1}{\log_2(j+1)} \textit{ ideal discounted cumulative gain}.$$

F@N è il *fallout* (ovvero $\frac{|fp|}{|fp|+|tn|}$) calcolato tenendo conto solo delle prime N raccomandazioni per ogni utente.

R@N è il recall calcolato tendo conto solo delle prime N raccomandazioni per ogni utente;

Average **P**recision: $\mathbf{AP}_u = \frac{1}{|I_r(u)|} \sum_{i \in I_r(u)} \frac{pr_i(u)}{i}$

Che rappresenta la precision calcolata e mediata solo su un numero di elementi raccomandati pari agli indici degli elementi rilevanti per l'utente u_u.

$\mathbf{TOPK}_u(k) = \frac{pr_k(u)}{|I_r(u)|}$, che rappresenta la percentuale di hit tra i primi k suggeriti all'utente u_u.

Con le grandezze appena introdotte si disponde di tutti gli strumenti per definire le misure:

- **M**ean **A**verage **P**recision: $\mathbf{MAP} = \frac{1}{|U_r|} \sum_{u \in U_r} AP_u$;
- **N**ormalized **D**iscounted **C**umulative **G**ain: $\mathbf{NDCG} = \frac{1}{|U_r|} \sum_{u \in U_r} \frac{DCG(u)}{IDCG(u)}$ [†];
- $\mathbf{NDCG@N} = \frac{1}{|U_r|} \sum_{u \in U_r} \frac{DCG@N(u)}{IDCG@N(u)}$, dove DCG@N e IDCG@N sono le stesse definite sopra, ma prendono in considerazione soltanto le prime N posizioni;
- **F1@N**, che rappresenta la F1-score della sezione 1.1, ma dove si tiene conto soltanto delle prime N posizioni;
- **A**rea **U**nder the **C**urve: **AUC** è definita come l'area sotto la curva ROC[‡], ovvero la curva che si ottiene disegnando i valori di **R@N** (recall) in funzione di **F@N** (fallout);

Le metriche F1@N e NDCG@N sono anche dette *metriche top*-N, perchè tengono conto solo delle prime N raccomandazioni per utente.

1.4 Comportamento delle misure quando A è unaria

In questa sezione si studia il significato delle misure sopra descritte applicate a BR, BD e $BN(\mathcal{N})$ nel caso in cui A sia una matrice unaria. Un riepilogo dei risultati ottenuti si può trovare in tabella 1.1.

[†]È da notare che questa misura tende a 1 (come dimostrato in [14]) per il numero di suggerimenti che tende all'infinito, quindi è una buona misura soltanto se non si è interessati alla scalabilità.

[‡]La curva ROC (**R**eceiver **O**perating **C**haracteristic) mostra le capacità di generalizzazione in un problema di classificazione binaria al variare della soglia. Tale curva è stata introdotta in Gran Bretagna durante la seconda guerra mondiale per migliorare l'efficacia dei radar. In particolare, era necessario capire quale fosse la soglia che consentisse sì di rimuovere il "rumore" (ad esempio uccelli in volo) limitando tuttavia il numero di falsi negativi, che avrebbero portato al mancato abbattimento di aerei nazisti, con la conseguente perdita di numerosi civili.

Successivamente, lo studio di tale curva si è rivelato interessante in campi completamente diversi, quali la radiologia [5].

1.4.1 Misure con A unaria e matrici di tipo $\mathtt{BR}$

In questa parte della trattazione si è scelto di paragrafare le considerazioni secondo la classificazione espressa pocanzi:

Misure di classificazione Si considerino le matrici ottenute come output di algoritmi a valori reali. Come già detto durante l'esposizione delle stesse, le *metriche di classificazione* non possono essere usate: la loro definizione richiede intrinsecamente che venga effettuata una qualche sogliatura. È richiesto infatti ricavare un insieme degli elementi suggeriti.

Misure di rating Per quanto riguarda le *metriche di rating*, non ci sono significativi cambiamenti nel caso unario: infatti, la formula $|a_{l,h} - b_{l,h}|$ si riduce ad essere $|b_{l,h}|$ se $(l,h) \in \overline{\Omega}$ e $|1 - b_{l,h}|$ altrimenti, ma essendo $\mathtt{BR}$ una matrice a coefficienti reali arbitrari questa riscrittura non altera l'interpretazione della metrica.

Misure di ranking Considerando le *metriche di ranking*, si nota che MAP e AUC non dipendono dai valori di A (eccetto che per la scelta della soglia, che nel caso unario sarà naturalmente $\sigma_A = 1$) e quindi non subiscono variazioni nel caso unario.

NDCG invece, poichè ottenuta a partire da DCG e IDCG, ha valore dipendente anche dagli elementi di A.

In particolare, **IDCG** si semplifica sensibilmente. Infatti, $a_{l,\Phi_l(j)} = 1$ per ogni j compreso tra 1 e $|I_r(l)|$, per cui tutti i numeratori diventano $2^1 - 1 = 1$ e la formula si riduce a

$$\mathbf{IDCG}(l) = \sum_{j=1}^{|I_r(l)|} \frac{1}{\log_2(j+1)}$$

Per quanto riguarda **DCG**, sia $J(l) = \{j \mid j \leqslant |I_r(l)| \text{ e } a_{l,\pi_l(j)} = 1\}$ l'insieme dei $j \leqslant |I_r(l)|$ per cui il j-esimo elemento raccomandato dall'algoritmo all'utente u_l è effettivamente interessante per l'utente stesso. Per ogni altro j, dato che $a_{l,\pi_l(j)} = 0$, il numeratore nell'espressione che definisce **DCG** diventa $2^0 - 1 = 0$, e dunque tale termine scompare dall'espressione. Dunque si ottiene

$$\mathbf{DCG}(l) = \sum_{j \in J(l)} \frac{1}{\log_2(j+1)}$$

Di conseguenza, il rapporto $\frac{\mathbf{DCG}(l)}{\mathbf{IDCG}(l)}$ diventa analogo al recall per l'utente u_l (che corrisponde a $\frac{|J(l)|}{|I_r(l)|}$), in cui però ogni elemento ottiene un peso che dipende dalla sua posizione nella lista di raccomandazioni: gli elementi più in alto nella lista sono più importanti.

NDCG corrisponde quindi alla media di questi "recall pesati" su ogni utente.

Misure top-N Riprendendo alcune considerazioni fatte pocanzi, per le *metriche top-N* si ha che F1@N non dipende dai valori di A, eccezion fatta per la scelta della soglia. Diversamente, NDCG@N subisce le stesse modifiche di NDCG, in quanto dipende da DCG e IDCG.

1.4.2 Misure con A unaria e matrici di tipo BD

Misure di clssificazione In questo caso, le *metriche di classificazione* hanno le stesse caratteristiche del caso in cui A non sia unaria (con soglia per A pari a $\sigma_A = 1$).

Misure di rating Considerazioni diverse valgono invece per le *metriche di rating*. In particolare, per quanto riguarda MAE, si ha che $|a_{l,h} - \tilde{b}_{l,h}|$ vale 0 se e solo se $a_{l,h} = \tilde{b}_{l,h}$, cioè se e solo se la coppia (l, h) corrisponde a un vero positivo o un vero negativo, e altrimenti vale 1. Quindi

$$\mathbf{MAE} = \frac{1}{|R|} \sum_{fp \cup fn} 1 = \frac{1}{|R|} \left(|R| - \sum_{tp \cup tn} 1 \right) = 1 - \frac{|tp| + |tn|}{|R|} = 1 - \mathbf{A}$$

Da questo si deduce che la metrica **MAE** su $\widetilde{B}$ è equivalente alla metrica Accuracy.

Analogamente, per RMSE, abbiamo che $|a_{l,h} - \tilde{b}_{l,h}|^2$ vale 0 se e solo se $a_{l,h} = \tilde{b}_{l,h}$, cioè se e solo se la coppia (l, h) corrisponde a un vero positivo o un vero negativo, e altrimenti vale 1. Quindi

$$\mathbf{RMSE} = \sqrt{\frac{1}{|R|} \sum_{fp \cup fn} 1} = \sqrt{\frac{1}{|R|} \left(|R| - \sum_{tp \cup tn} 1 \right)} = \sqrt{1 - \frac{|tp| + |tn|}{|R|}} = \sqrt{1 - \mathbf{A}}$$

Dunque vale $\mathbf{MAE} = \mathbf{RMSE}^2$, e quindi l'ordine tra le due è lo stesso. Di conseguenza anche **RMSE** è equivalente ad Accuracy.

Misure di ranking Le *metriche di ranking*, invece, perdono completamente di significato, poichè richiedono di conoscere l'ordine tra gli elementi per un certo utente, ma, su matrici unarie, questo ordine non è definito. Più nel dettaglio, è ragionevole richiedere, per questa combinazione di matrici A e BD, che una metrica soddisfi la seguente proprietà 1.4.1:

Proprietà 1.4.1. *[Invarianza unaria] Una misura è **invariante unaria** se non dipende dall'ordine in cui sono considerati gli elementi sulle righe della matrice.*

In termini più formali, una misura $\mu : \mathcal{M}(n, k, \mathbb{R}) \to \mathbb{R}$ *si dice* ***invariante unaria*** *se,*

$$\forall A = \begin{pmatrix} A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_i \\ \cdot \\ \cdot \\ \cdot \\ A_{n-1} \\ A_n \end{pmatrix}, \ \forall \pi \in \mathcal{S}_n \text{ t.c. } P_{Ai} = \begin{pmatrix} A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ \pi(A_i) \\ \cdot \\ \cdot \\ \cdot \\ A_{n-1} \\ A_n \end{pmatrix} \mu(A) = \mu(P_{Ai})$$

Si osserva che, per le metriche MAP, NDCG e AUC su matrici di tipo BD, la Teorema 1.4.1 non è valida.

Per confermare questa intuzione, si dimostra che il valore di AP_l e di $DCG(l)$ cambia al variare dell'ordine degli elementi in alcuni casi particolari:

MAP - Senza perdere di generalità, si consideri $v \in \{0,1\}^3$, riga di A ed una riga di BD: $w \in \{0,1\}^3$.

$$v = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \qquad\qquad w = \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}$$

Si osserva che sono possibili due permutazioni:

$$\pi_{l_1}(i) = \begin{cases} 3 & \text{se i = 1} \\ 1 & \text{se i = 2} \\ 2 & \text{se i = 3} \end{cases} \text{ e } \pi_{l_2}(i) = \begin{cases} 3 & \text{se i = 1} \\ 2 & \text{se i = 2} \\ 1 & \text{se i = 3} \end{cases}$$

Si valuta soltanto AP_l, in quanto MAP non è che la media sugli utenti delle varie AP_l.

Mediante la permutazione π_{l_1} $pr_1(l) = 1$ e $pr_2(l) = 1$, quindi

$$AP_l = \frac{1}{|I_r(l)|} \sum_{i \in I_r(l)} \frac{pr_i(l)}{i} = \frac{1}{|I_r(l)|} \cdot \left(\frac{pr_1(l)}{1} + \frac{pr_2(l)}{2} \right) = \frac{1}{2} \left(\frac{1}{1} + \frac{1}{2} \right) = \frac{3}{4}$$

ma, con la seconda permutazione, $pr_1(l) = 0$ e $pr_2(l) = 1$ e quindi

$$AP_l = \frac{1}{|I_r(l)|} \sum_{i \in I_r(l)} \frac{pr_i(l)}{i} = \frac{1}{|I_r(l)|} \cdot \left(\frac{pr_1(l)}{1} + \frac{pr_2(l)}{2} \right) = \frac{1}{2} \left(\frac{0}{1} + \frac{1}{2} \right) = \frac{1}{4}$$

Tali valori di AP sono diversi, quindi AP_l (e quindi MAP) non è adeguata a matrici di tipo BD

NDCG - Senza perdere di generalità, si consideri $v \in \{0,1\}^4$, riga di A ed una riga di BD: $w \in \{0,1\}^4$.

$$v = \begin{pmatrix} 0 & 1 & 0 & 1 \end{pmatrix} \qquad\qquad w = \begin{pmatrix} 1 & 0 & 1 & 1 \end{pmatrix}$$

Due permutazioni possibili sono:

$$\pi_{l_1}(i) = \begin{cases} 1 & \text{se i = 1} \\ 4 & \text{se i = 2} \\ 3 & \text{se i = 3} \\ 3 & \text{se i = 4} \end{cases} \text{ e } \pi_{l_2}(i) = \begin{cases} 2 & \text{se i = 1} \\ 4 & \text{se i = 2} \\ 1 & \text{se i = 3} \\ 3 & \text{se i = 4} \end{cases}$$

Usando la prima permutazione:

$$DCG_l = \sum_{j=1}^{2} \frac{2^{a_{l,\pi_{l_1}(j)}} - 1}{\log_2(j+1)} = \frac{2^{a_{l,\pi_{l_1}(1)}} - 1}{\log_2(1+1)} + \frac{2^{a_{l,\pi_{l_1}(2)}} - 1}{\log_2(2+1)} = \frac{2^{a_{l,1}} - 1}{\log_2(2)} + \frac{2^{a_{l,4}} - 1}{\log_2(3)} =$$

$$\frac{0}{1} + \frac{1}{\log_2(3)} = \frac{1}{\log_2(3)}$$

Usando invece π_{l_2}:

$$DCG_l = \sum_{j=1}^{2} \frac{2^{a_{l,\pi_{l_2}(j)}} - 1}{\log_2(j+1)} = \frac{2^{a_{l,\pi_{l_2}(1)}} - 1}{\log_2(1+1)} + \frac{2^{a_{l,\pi_{l_2}(2)}} - 1}{\log_2(2+1)} = \frac{2^{a_{l,2}} - 1}{\log_2(2)} + \frac{2^{a_{l,4}} - 1}{\log_2(3)} =$$

$$\frac{1}{1} + \frac{1}{\log_2(3)}$$

che sono chiaramente diversi.

AUC - Senza perdere di generalità, si consideri $v \in \{0,1\}^{30}$, riga di A ed una riga di BD: $w \in \{0,1\}^{30}$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$w =$	1	0	1	0	0	0	1	0	1	1	0	1	1	0	1	1	1	1	0	0	1	0	0	1	1	1	0	1	1	0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$v =$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Si scelgono due diverse permutazioni $\pi_{l_1} \neq \pi_{l_2}$ che inducono il medesimo ordinamento w_s a partire da w.

	15	17	16	26	1	3	12	24	10	28	21	18	7	29	13	9	25	27	23	19	14	5	8	11	20	4	30	22	2	6
$w_s =$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

$$
\begin{array}{rl}
 & \begin{array}{cccccccccccccccccccccccccccc} 1 & 3 & 12 & 10 & 15 & 17 & 16 & 26 & 24 & 28 & 21 & 18 & 7 & 29 & 13 & 9 & 25 & 27 & 23 & 19 & 14 & 5 & 8 & 11 & 20 & 4 & 30 & 22 & 2 & 6 \end{array} \\
w_s = & \left(\begin{array}{cccccccccccccccccccccccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)
\end{array}
$$

Queste due permutazioni che danno origine a w_s a partire da w portano a due diverse curve ROC e due diverse aree ad esse sottese, come mostrato dalla figura 1.1. Questo dimostra che non è possibile utilizzare la misura AUC, poichè non è stabile per BD fissata.

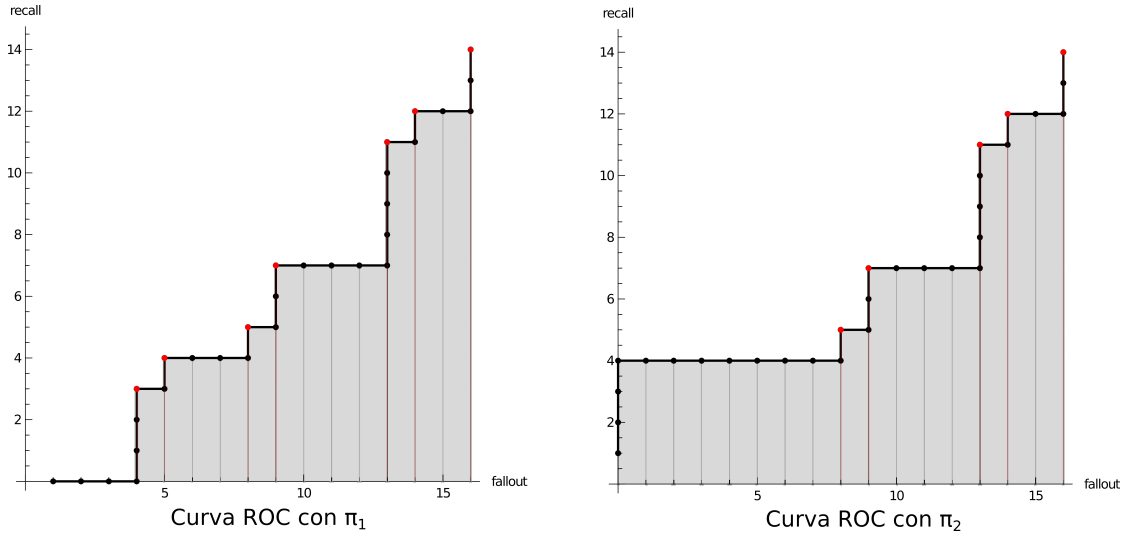

Figura 1.1: In questa figura si nota che la permutazione su una riga della matrice B porta a due diverse curve ROC, con conseguenti diversi integrali.

1.4.3 Misure con A unaria e matrici di tipo $\mathsf{BN}(\mathcal{N})$

Misure di classificazione In questa ipotesi le *metriche di classificazione* hanno la seguente peculiarità: dato che ad ogni utente sono forniti esattamente N suggerimenti, F1 coincide con la relativa metrica top-N (F1@N).

Misure di rating Dato che il caso $\mathsf{BN}(\mathcal{N})$ è un caso particolare del caso BD, valgono tutte le considerazioni espresse nei casi precedenti, ossia le due metriche MAE e RMSE ottengono gli stessi risultati dell'accuracy.

Misure di ranking Anche per questa categoria di misure vale il ragionamento espresso sopra, pertanto le tre misure che non sono invarianti unarie non sono una selta ragionevole.

Misure top-N Questo particolare tipo di matrice B è l'unico caso in cui ha senso approfondire le *metriche top-N*. Più nel dettaglio, F1@N può comunque essere calcolata anche in questo caso, poichè non è rilevante l'ordine dei suggerimenti ma si è interessati soltanto a quali sono i primi N (che in questo caso sono anche gli unici N).

Tale ragionamento non si applica invece a NDCG@N, che richiede di conoscere anche l'ordine dei primi N suggerimenti.

	BR	**BD**	**BN(N)**
misure di classificazione			
F1	—	$F1(BD)$	$F1(BN(\mathcal{N}))$
A	—	$A(BD)$	$A(BN(\mathcal{N}))$
misure di rating			
MAE	$MAE(BR)$	$1 - A(BD)$	$1 - A(BN(\mathcal{N}))$
RMSE	$MAE(BR)$	$\sqrt{1 - A(BD)}$	$\sqrt{1 - A(BN(\mathcal{N}))}$
misure di ranking			
MAP	$MAP(BR)$	—	$MAP(BN(\mathcal{N}))$
AUC	$AUC(BR)$	—	$AUC(BN(\mathcal{N}))$
NDCG	$NDCG(BR)$	—	—
*misure top-*N			
NDCG@N	$NDCG@N(BR)$	—	—
F1@N	$F1@N(BR)$	—	$F1(BN(\mathcal{N}))$

Tabella 1.1: Tabella riassuntiva delle metriche con vari tipi di matrici.
Legenda:

- Se in una casella appare la combinazione di misura e tipo di matrice corrispondenti alla riga e alla colonna, sta a significare che tale misura può essere calcolata in quella situazione e non coincide con altre;
- Un trattino (-) indica che tale misura non può essere calcolata;
- La presenza nella cella di un'altra misura indica che la valutazine offerta dalle due metriche su quel caso è la stessa.

Capitolo 2

Cenni sui metodi risolutivi

In questo capitolo si trattano alcuni algoritmi basati su fattorizzazione di matrici e arcinoti in letteratura per il problema in cui sono noti i rating, nondimeno vengono approfondite alcune difficoltà con cui questi algoritmi si scontrano quando usati su dati unari. Nonostante tali algoritmi offrano un panorama piuttosto chiaro degli approcci più usati in questo contesto, non sono che un campione di quello che si trova in letteratura e che è stato studiato in [2, 15].

2.1 PSVD

Questo approccio è basato sulla fattorizzazione SVD pura, definita come segue.

Definizione 2.1.1 (SVD)**.** *Data una matrice* $A \in M(n, m, \mathbb{R})$ *con* $n < m$*, si dice che l'espressione*

$$S\Sigma U^T$$

è l'SVD di A *se* Σ *è una matrice diagonale di taglia* n *con elementi non-negativi, disposti in ordine decrescente,* S *è una matrice ortogonale* * *di taglia* n*,* U *è una matrice ortogonale di taglia* m *e inoltre* $A = S\Sigma U^T$*. Gli elementi di* Σ *si chiamano* valori singolari *di* A*.*

È noto ([1]) che considerare i soli k valori singolari più grandi dell'SVD di una matrice A fornisce la migliore approssimazione di A come prodotto di due matrici di rango al massimo k, ovvero le due matrici H e W di rango k che minimizzano

$$\|A - WH^T\|_F$$

Tale approssimazione si ottiene a partire dall'SVD come segue: sia Σ_k la matrice che si ottiene da Σ considerando solo i primi k elementi sulla diagonale, e sia $\overline{\Sigma}_k$ la matrice tale che $\overline{\Sigma}_k{}^2 = \Sigma_k$ (che quindi è una matrice diagonale che ha per elementi le radici quadrate dei k più grandi valori singolari di A). Allora vale che $A_k = (S\overline{\Sigma}_k)(\overline{\Sigma}_k U^T)$ è una matrice di rango k ed è quella che minimizza $\|A - A_k\|$ tra tutte le matrici di rango

*Una matrice $M \in M(n, \mathbb{R})$ si dice *ortogonale* se è tale che $M^t M = MM^t = I_n$

k. Quindi scegliere $W = S\overline{\Sigma}_k$ e $H = U\overline{\Sigma}_k$ [†] produce la fattorizzazione desiderata (infatti tali W e H hanno rango al massimo k, essendo il rango di Σ_k al più k).

L'algoritmo PSVD in realtà non calcola l'intera fattorizzazione SVD della matrice, per poi scartare i valori singolari bassi, perchè tale procedimento sarebbe computazionalmente dispendioso. Procede invece al calcolo diretto delle matrici W e H per un k fissato, ottenendo una complessità di calcolo minore.

Il resto di questa sezione si occupa di dimostrare l'esistenza dell'SVD.

2.1.1 Esistenza della Singolar Value Decompositon

Si fornisce in questa parte una dimostrazione dell'esistenza della fattorizzazione. Definiamo $C = AA^T$ e $D = A^TA$. Da queste definizioni segue immediatamente che:

Lemma 2.1.1. C *e* D *sono simmetriche.*

Dimostrazione. Dalle proprietà della trasposta, si ha $C^T = (AA^t)^t = AA^t = C$. Analogamente si dimostra che D è simmetrica. □

Di conseguenza, valgono le ipotesi per applicare a C e a D il teorema spettrale:

Teorema 2.1.2 (Teorema spettrale)**.** *Sia* $P \in S(n, \mathbb{R})$*, allora esistono* $x_1, x_2, \ldots, x_n$ *autovettori ortonormali di* P*, con autovalori* $\lambda_1, \lambda_2, \ldots, \lambda_n$ *reali.*

Inoltre, per come sono definite C e D segue che

Lemma 2.1.3. *Sia* C *che* D *sono semi-definite positive.*

Dimostrazione. Dimostriamo la tesi per D: la dimostrazione per C è analoga, sostituendo A con A^T. La tesi equivale a $x^tA^tAx \geqslant 0$, ma $x^tA^tAx = (Ax)^t(Ax) \geqslant 0$, perché il prodotto scalare standard è definito positivo. □

Corollario 2.1.4. *Sia* C *che* D *hanno autovalori reali non negativi.*

Tali autovalori sono dunque quadrati di numeri reali non negativi, ordinati in senso decrescente $\sigma_1^2 \geqslant \sigma_1^2 \geqslant \cdots \geqslant \sigma_d^2$, tali che $\sigma_1, \sigma_2, \ldots, \sigma_r \in \mathbb{R}^+$ e $\sigma_{r+1}, \ldots, \sigma_d = 0$.

Quindi sia

$$U = \left(\begin{array}{c|c|c|c} & & & \\ x_1 & x_2 & \cdots & x_r \\ & & & \end{array} \right) \in M(m, r, \mathbb{R})$$

la matrice che ha per colonne gli autovettori ortonormali di D relativi ad autovalori positivi.

Siano $y_i = \frac{1}{\sigma_i} Ax_i \ \forall i = 1, \ldots, r$, allora vale il seguente lemma:

[†] Si osservi che $\overline{\Sigma}_k = \overline{\Sigma}_k^t$, poichè $\overline{\Sigma}_k$ è diagonale.

Lemma 2.1.5. $\forall i \in \{1, \dots, r\}$, y_i *sono autovettori ortonormali per* C.

Dimostrazione. Dimostriamo innanzitutto che gli y_i sono autovettori di C. È possibile riscrivere la tesi come $Cy_i = \lambda_i y_i$, ovvero $AA^t y_i = \lambda_i y_i$.

Si ha

$$AA^t y_i = AA^t \left(\frac{1}{\sigma_i} A x_i\right) = A \left(\frac{1}{\sigma_i} A^t A x_i\right) = A \left(\frac{1}{\sigma_i} \sigma_i^2 x_i\right) = \sigma_i^2 \frac{1}{\sigma_i} A x_i = \sigma_i^2 y_i$$

che corrisponde alla prima parte della tesi scegliendo $\lambda_i = \sigma_i^2$.

Rimane da dimostrare che sono ortonormali, cosa che segue dalla seguente catena di uguaglianze

$$\begin{aligned} y_i^t y_j &= \left(\frac{1}{\sigma_i} A x_i\right)^t \frac{1}{\sigma_j} A x_j \\ &= \frac{1}{\sigma_i \sigma_j} x_i^t A^t A x_j \\ &= \frac{1}{\sigma_i \sigma_j} x_i^T B x_j \\ &= \frac{1}{\sigma_i \sigma_j} x_i^T \sigma_j^2 x_j \\ &= \frac{\sigma_j}{\sigma_i} x_i^T x_j \end{aligned}$$

Quindi, poichè x_i e x_j sono ortonormali, si ha la tesi. □

Sia

$$S = \left(\begin{array}{c|c|c|c} & & & \\ y_1 & y_2 & \cdots & y_r \\ & & & \end{array} \right) \in M(n, r, \mathbb{R})$$

la matrice che ha per colonne gli autovettori ortonormali relativi ad autovalori non nulli di C e si consideri la matrice $\Sigma = S^T A U$. Un suo generico elemento (i, j) vale $(S^T A U)_{ij} = y_j^T A x_i = y_j^T \sigma_i y_i = \sigma_i y_j^T y_i$, quindi, poichè gli y_i sono ortonormali, tale matrice è diagonale con elementi $\sigma_1 \dots \sigma_r$.

Inoltre, poichè S e U hanno per colonne vettori ortonormali, $SS^T = I_n$ e $UU^T = I_m$, quindi è possibile moltiplicare l'uguaglianza $S^T A U = \Sigma$ a sinistra per S e a destra per U^T, ottenendo il seguente teorema:

Teorema 2.1.6. *Sia* $A \in M(n, m, \mathbb{R})$ *e siano* $C = MM^T$, $D = M^T M$, $U \in M(d, r, \mathbb{R})$ *matrice che ha per colonne gli autovettori ortonormali relativi ad autovalori non nulli di* D *e* $S \in M(p, r, \mathbb{R})$ *matrice che ha per colonne gli autovettori ortonormali relativi ad autovalori*

non nulli di C. *Allora la matrice* $\Sigma = S^T A U$ *è diagonale e ha per elementi le radici quadrate positive degli autovalori della matrice* D, *ossia*

$$S^T A U = \Sigma = \begin{pmatrix} \sigma_1 & & & & 0 \\ & \sigma_2 & & & \\ & & \ddots & & \\ & & & \sigma_{r-1} & \\ 0 & & & & \sigma_r \end{pmatrix}$$

Inoltre vale che $A = S\Sigma U^T$.

2.2 NMF

La fattorizzazione **N**onnegative**M**atrix**F**actorization (**NMF**) si pone come obiettivo quello di trovare, a partire da una matrice $A \in M(n, m, \mathbb{R})$ le matrici $W \in M(n, k, \mathbb{R}^+)$ e $H \in M(m, k, \mathbb{R}^+)$ che minimizzano

$$\|A - WH^T\|_F$$

Nel caso dei recommendation system, si prende anche in considerazione il problema di minimizzare $\|A_\Omega - (WH^T)_\Omega\|_F$.

Il problema di questa funzione da minimizzare è che non è convessa, ed esistono quindi numerosi minimi locali. L'algoritmo di risoluzione più conosciuto è il cosiddetto *metodo dei minimi quadrati alternati* (**ANLS**), che sfrutta la seguente osservazione: considerando la matrice W come costante, il problema di ottimizzazione ottenuto è effettivamente convesso (e analogamente considerando fissa H). L'algoritmo dei minimi quadrati alternati procede dunque nel seguente modo:

1. Si sceglie a caso una matrice positiva W;
2. Si trova la matrice positiva H che minimizza $\|A - WH^T\|_F$;
3. Si trova la matrice positiva W che minimizza $\|A - WH^T\|_F$;
4. Si ripete dallo step 2 fino a soddisfare una condizione di errore.

In realtà, negli step 2 e 3 non è necessario trovare il minimo esatto, perchè già si sa che la matrice verrà successivamente modificata. Questo porta ad algoritmi più efficienti in pratica, per quanto la loro convergenza non sia dimostrata.

2.3 RNMF

RNMF è una variante dell'algoritmo NMF che aggiunge un termine di regolarizzazione alla funzione da minimizzare, che quindi diventa

$$\|A - WH^T\|_F + \lambda\|W\|_1 + \lambda\|H\|_1$$

dove $\lambda > 0$ è una costante fissata.

È necessario precisare che il grande vantaggio di questo metodo è che tende a favorire la presenza di zeri nelle matrici H e W, riducendo quindi il fenomeno dell'**overfitting**. In particolare, tale fenomeno può essere osservato facilmente analizzando i massimi valori reali che vengono calcolati dall'algoritmo. L'esempio più emblematico è quello dell'interpolazione polinomiale: se si sceglie di approssimare la funzione obiettivo con un polinomio di alto ordine si ottiene una funzione che vale esattamente quanto dovrebbe valere nei punti di cui si conosce il valore, ma approssima malissimo altrove, dove associa agli input output con valore elevatissimo. Nel nostro caso, l'overfitting si può presentare quando la matrice B vale 1 in ogni punto in cui A valeva 1, ma gli altri valori hanno un modulo molto grande.

Concludendo, bloccare tali valori mediante una regolarizzazione garantisce che non esplodano, ma è ancora più importante nel caso unario, dove è possibile ottenere la soluzione banale, ovvero matrice di tutti 1, che è anch'essa una forma di overfitting. Nella sezione seguente si studieranno i dettagli degli algoritmi applicati al caso unario.

È da notare che questa modifica è praticamente gratuita dal punto di vista della complessità computazionale, dunque esistono algoritmi altrettanto efficienti rispetto a quelli conosciuti per risolvere NMF.

2.4 Comportamento degli algoritmi nel caso unario

Come osservato nell'introduzione, il comportamento di alcuni di questi algoritmi è poco adeguato al caso unario. Infatti, consideriamo le matrici W e H definite da

$$W = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d \qquad\qquad H = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^d$$

È chiaro che entrambe queste matrici sono a termini non-negativi e hanno rango 1, quindi soddisfano le condizioni necessarie per gli algoritmi NMF e PSVD. Inoltre, si ha

$$(WH^T)_{i,j} = 1 \cdot 1 = 1 \quad \forall i, j \in \{1, 2, \dots d\}$$

e, quindi, $A_\Omega = (WH^T)_\Omega$: di conseguenza

$$\|A_\Omega - (WH^T)_\Omega\|_F = 0$$

e quindi W e H producono una matrice B che è ottima per le varianti di NMF e PSVD che tengono conto dei soli valori noti, ma che non ha alcun potere predittivo perché sostiene che a ogni utente piacerà ogni item.

Questo inconveniente si può risolvere con tecniche di *regolarizzazione*, cioè tecniche che modificano la funzione obiettivo in modo da penalizzare soluzioni che presentano un numero eccessivo di cifre 1 o una norma troppo elevata. Ad esempio, RNMF è un esempio di problema in cui è aggiunto a MNF un termine di regolarizzazione.

Un modo di modificare PSVD aggiungendo un termine di regolarizzazione nel caso di dati unari è cercare due matrici W e H che minimizzino la seguente espressione

$$\|A_\Omega - (WH^T)_\Omega\|_F + \lambda\|W\|_1 + \lambda\|H\|_1$$

con $\lambda > 0$ costante fissata, come per RMNF.

L'intuizione dietro a questa formula è di penalizzare le soluzioni che forniscono molti suggerimenti, preferendo meno consigli, ma più mirati.

2.5 Scelta della soglia per discretizzare i risultati

Si analizzano in questa parte alcune strategie per scegliere la soglia di discretizzazione delle matrici BR. Questa scelta di soglia si può fare sia con criteri che dipendono dall'algoritmo utilizzato che con criteri che dipendono dai risultati stessi. Questi ultimi si possono dividere in due famiglie principali: i metodi *globali* e quelli *per-utente*.

Un'altra categorizzazione è quella che concerne quando effettuare la sogliatura: in questo ambito si può parlare di sogliatura *a priori* e sogliatura *a posteriori*.

Un esempio di sogliatura a priori è il seguente: un algoritmo basato su SVD potrebbe produrre una matrice BR che contiene sia valori positivi che negativi: una scelta di discretizzazione potrebbe quindi essere quella di produrre una matrice BD con $\tilde{b}_{l,h} = 1$ se e solo se $b_{l,h} \geqslant 0$ e 0 altrimenti. D'altra parte questa scelta non può funzionare con metodi basati su NMF, dato che in tal caso tutti i valori di BR sono non-negativi.

Si nota che su matrici di 0 e 1 $\|A\|_1$ rappresenta il numero di caselle di valore 1 nella matrice A. Per quanto riguarda i metodi adattivi (o sogliatura a posteriori), si possono seguire diverse strategie:

- **Cercare** BD **con densità** λ**:** per farlo, si cerca una soglia σ per cui

  $$\frac{\|BD\|_1}{nm} \approx \lambda$$

  Per esempio, una scelta di $\lambda = 0.5$ equivale a scegliere come soglia la mediana di tutti i valori in BR: in tal caso circa il 50% dei valori risulta superiore alla soglia.

  Inoltre, sarebbe interessante dal punto di vista algoritmico studiare come determinare σ in funzione di λ, ad esempio si potrebbe individuare σ con un approcio teorico, oppure in modo adattivo, ossia mentre l'algoritmo sta girando, si valuta la densità: in caso tale densità sia superiore a λ, δ viene aumentata, altrimenti diminuita. In questo modo, a convergenza, si ottiene il σ giusto.

- **Cercare di aggiungere suggerimenti con densità simile ad A:** per farlo, si cerca una soglia σ per cui
$$\frac{\|BD_{\overline{\Omega}}\|_1}{|\overline{\Omega}|} \approx \frac{\|A\|_1}{nm}$$
L'intuizione dietro a questa scelta è di cercare di suggerire in tutto una quantità di elementi proporzionale a quella su cui si ha informazione.

- **Cercare di mantenere la percentuale di elementi diversi da A costante:** per fare questo, si cerca di imporre che vi siano più o meno tanti zeri in BD_Ω, in percentuale, quanti uni in $BD_{\overline{\Omega}}$. In formule:
$$\frac{\|BD_{\overline{\Omega}}\|_1}{|\overline{\Omega}|} \approx 1 - \frac{\|BD_\Omega\|_1}{|\Omega|}$$

- **Suggerire a ogni utente un numero di elementi proporzionale a quanti ne ha graditi:** per ottenere questo risultato, bisogna ovviamente scegliere una soglia per ogni utente. Si sceglierà quindi una soglia σ_l per cui
$$|\{h : b_{l,h} \geqslant \sigma_l\}| = \lambda\,|\{h : a_{l,h} = 1\}|$$
e scegliendo quindi $\tilde{b}_{l,h} = 1$ se e solo se $b_{l,h} \geqslant \sigma_l$.

- **Suggerire a ogni utente esattamente N elementi:** analogamente a prima, si ottiene questo risultato con un σ_l tale che
$$|\{h : b_{l,h} \geqslant \sigma_l\}| = N$$
In tal caso, si ottiene chiaramente una matrice di tipo $BN(\mathcal{N})$.

Capitolo 3

Sperimentazione sulle misure

Per valutare l'*efficacia* delle varie metriche e determinare in quale rapporto stanno tra loro si è scelto di applicarle a matrici create *ad hoc*, così da studiarne il comportamento in casi significativi. Per preparare il terreno alla ricerca futura si forniscono alcuni dataset, approfonditi in Appendice B.

Prima di approfondire il lato sperimentale si premette una considerazione: è interessante studiare il comportamento di queste metriche, poichè nel caso in cui siano tutte equivalenti allora sarebbe inutile utilizzarne più di una; sarebbe altresì sufficiente scegliere un *rappresentante* della classe. Diversamente, nel caso in cui le misure non fossero coerenti, sarebbe importante capire quali misure selezionano una certa caratteristica e quali un'altra, in modo da associare ad ogni misura il miglior campo di utilizzo.

Sviluppare una simile classificazione tra metriche semplifica di molto il lavoro della ricerca futura, in quanto di fronte ad un nuovo algoritmo si può scegliere quali metriche usare per analizzarne i risultati.

Prima di valutare le misure sui dati, è necessario rivedere le definizioni delle misure al fine di ottenere 1 come risultato quando la soluzione approssimata **coincide** con quella esatta, 0 quando l'approssimazione è pessima.

Questo è il comportamento di Accuracy, F1, MAP, AUC e NDCG, ma non è così per:

MAE: $[0, +\infty[$, dove 0 si ottiene quando $B = A$. Nelle sperimentazioni si è scelto di utilizzare **C**orrected**MAE** (**CMAE**) $= \frac{1}{1+MAE}$, che invece ha il comportamento standard;

RMSE: $[0, +\infty[$, con la stessa semantica di MAE. Si definisce di conseguenza **C**orrected**RMSE** (**CRMSE**) $= \frac{1}{1+RMSE}$.

Gli esperimenti condotti sulle misure hanno questa struttura:

1. Generazione di soluzioni "esatte" di dimensione n. Senza perdere di generalità, si scelgono come soluzioni esatte matrici triangolari inferiori della forma

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \cdot & & 1 & & & & \cdot \\ \cdot & & & 1 & & & \cdot \\ & & & & \ddots & & \\ \cdot & & & & & & \cdot \\ 1 & 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & 1 & \cdots & 1 & 1 \end{pmatrix}$$

2. Generazione di soluzioni approssimate BR a valori casuali nell'intervallo $[0, 1]$. Queste matrici BR vengono sogliate nei casi in cui la metrica ne richieda la sogliatura (F_1 e Accuracy). Si è scelto di non utilizzare matrici di tipo BD, poichè le uniche metriche definite in tali casi sono F_1 e Accuracy (come si vede nella tabella 1.1) e per tali metriche le matrici BR vengono sogliate, divenendo di fatto del tipo BD;

3. Generazione di soluzioni approssimate BR a valori casuali nell'intervallo $[0, 1]$. Questo secondo tipo utilizza la distribuzione di probabilità triangolare:

$$P(x; l, m, r) = \begin{cases} \frac{2(x-l)}{(r-l)(m-l)} & \text{per } l \leqslant x \leqslant m, \\ \frac{2(r-x)}{(r-l)(r-m)} & \text{per } m \leqslant x \leqslant r, \\ 0 & \text{altrimenti.} \end{cases}$$

   Con $l = 0$, $r = 1$ e $m = 1$, ovvero la distribuzione di probabilità è sblianciata verso 1.

   Anche per questo tipo di matrici vale che la considerazione fatta sopra, ovvero che non ha senso ripetere gli esperimenti per matrici di tipo BD;

4. Applicazione di funzioni di correlazione (Kendall τ e Spearman ρ), così da ottenere una matrice di correlazione tra le misure.

   L'**indice di correlazione per ranghi** (misura di correlazione Spearman-rho) tra due vettori X e Y è definito come il rapporto tra la covarianza tra X e Y ordinati in senso decrescente e il prodotto delle deviazioni standard di X e Y. In formula,

$$\rho_{rg_X, rg_Y} = \frac{\operatorname{cov}(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}}$$

   Il **coefficiente di correlazione per rango di Kendall** di una coppia di vettori X e Y è definito come segue:

$$\tau = \frac{2}{n(n-1)} \sum_{i<j} \operatorname{sgn}(x_i - x_j) \operatorname{sgn}(y_i - y_j)$$

dove n è la dimensione dei vettori X e Y;

5. Studio del comportamento di tali correlazioni al crescere del numero di soluzioni approssimate, di utenti, di oggetti e di entrambi.

Nota: Si osservi che un valore di correlazione molto alto (vicino a 100) significa che le misure sono più o meno equivalenti.

Si è osservato sperimentalmente che le misure definite nel Capitolo 1 sono maggiormente correlate nei casi in cui la matrice ha la forma di un "rettangolo basso e largo", in termini più formali $n << m$, dove n rappresenta il numero di utenti ed m il numero di articoli.

Si può asserire infatti che per piattaforme in cui il numero di utenti sia inferiore di ordini di grandezza rispetto al numero di contenuti, le misure del gruppo *classificazione* e *rating* (ovvero CMAE, CRMSE, F1 e Accuracy) sono pressochè equivalenti.

Si osservino, a sostegno di questa tesi i valori contenuti nelle tabelle 3.1 e 3.2.

	CMAE	**CRMSE**	**F_1**	**ACC**	**MAP**	**AUC**	**NDCG**
CMAE	100.0	99.0	95.7	95.7	4.7	6.0	4.4
CRMSE	99.0	100.0	95.7	95.8	5.1	6.4	4.8
F_1	95.7	95.7	100.0	100.0	2.6	4.1	3.0
ACC	95.7	95.8	100.0	100.0	3.2	5.0	3.5
MAP	4.7	5.1	2.6	3.2	100.0	52.8	60.7
AUC	6.0	6.4	4.1	5.0	52.8	100.0	43.5
NDCG	4.4	4.8	3.0	3.5	60.7	43.5	100.0

Tabella 3.1: Misura di correlazione Spearman-rho su 10000 matrici approssimate BR, con A e BR di dimensione 32×1024

	CMAE	**CRMSE**	**F_1**	**ACC**	**MAP**	**AUC**	**NDCG**
CMAE	100.0	91.2	81.8	82.0	3.1	4.0	3.0
CRMSE	91.2	100.0	81.9	82.0	3.4	4.3	3.2
F_1	81.8	81.9	100.0	98.5	1.7	2.8	2.0
ACC	82.0	82.0	98.5	100.0	2.1	3.3	2.4
MAP	3.1	3.4	1.7	2.1	100.0	36.8	43.1
AUC	4.0	4.3	2.8	3.3	36.8	100.0	29.8
NDCG	3.0	3.2	2.0	2.4	43.1	29.8	100.0

Tabella 3.2: Misura di correlazione Kendall-tau su 10000 matrici approssimate BR, con A e BR di dimensione 32×1024

Considerazioni diverse valgono per MAP, AUC e NDCG, le *misure di ranking* che sono poco correlate tra loro e per nulla con gli altri due gruppi.

L'andamento delle correlazioni all'aumentare del numero di oggetti, tenendo fisso il numero di utenti è visibile anche nei seguenti grafici.

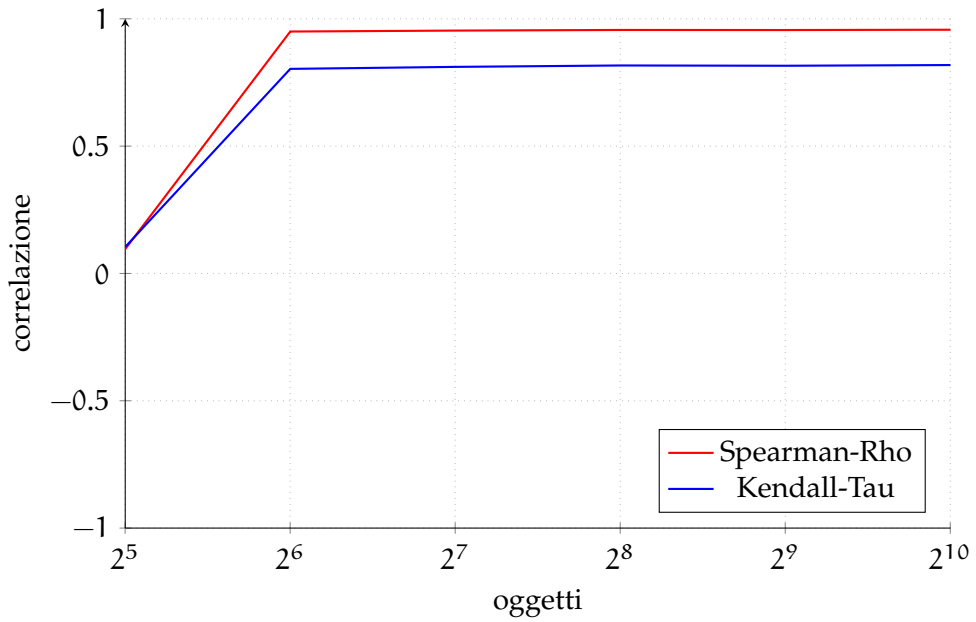

Figura 3.1: Correlazione tra F1 e CMAE, per un numero di utenti pari a 32, all'aumentare del numero di oggetti, per 10000 soluzioni approssimate.

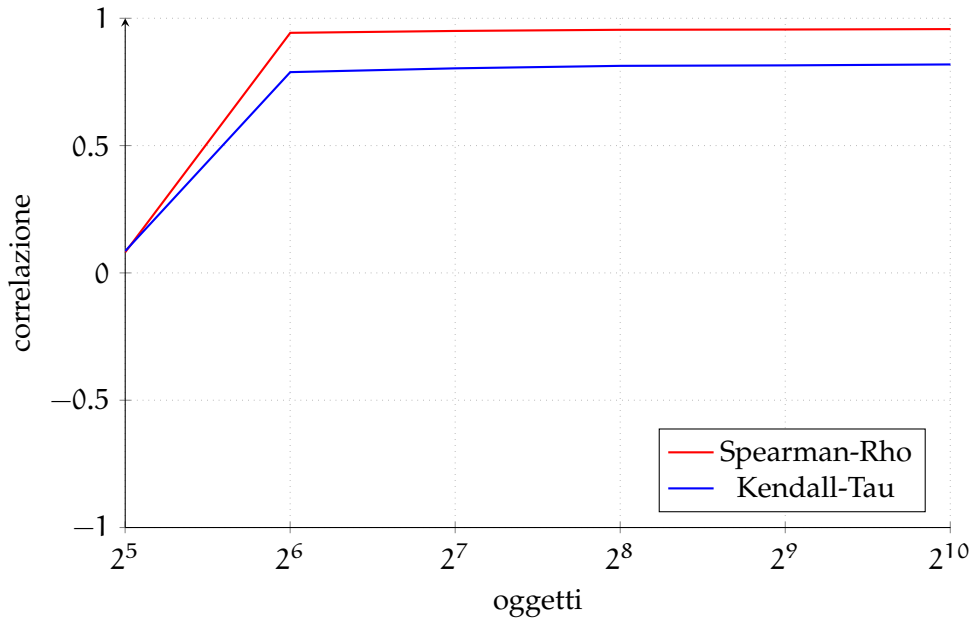

Figura 3.2: Correlazione tra F1 e CRMSE, per un numero di utenti pari a 32, all'aumentare del numero di oggetti, per 10000 soluzioni approssimate.

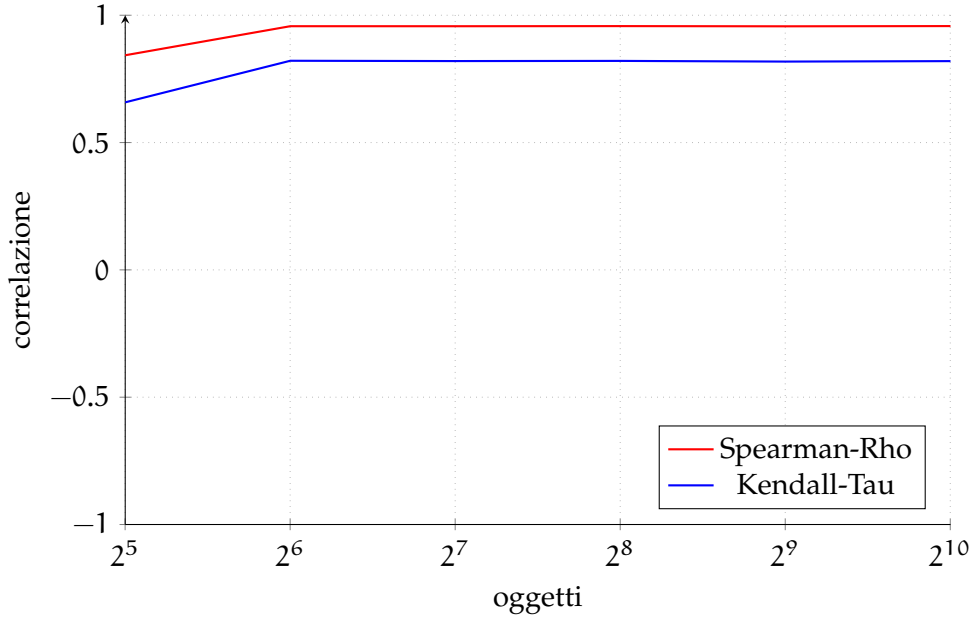

Figura 3.3: Correlazione tra Accuracy e CMAE, per un numero di utenti pari a 32, all'aumentare del numero di oggetti, per 10000 soluzioni approssimate.

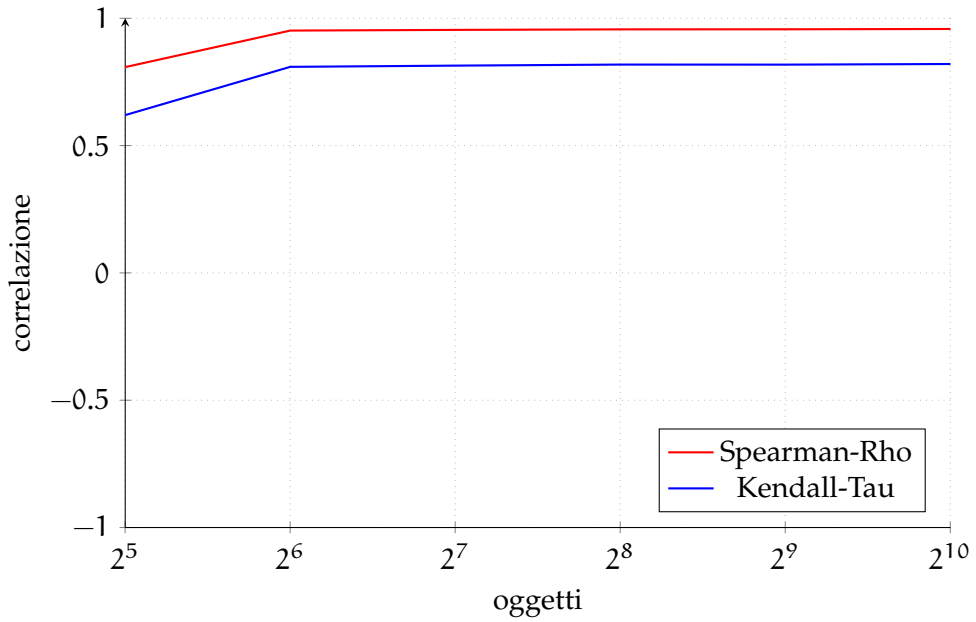

Figura 3.4: Correlazione tra Accuracy e CRMSE, per un numero di utenti pari a 32, all'aumentare del numero di oggetti, per 10000 soluzioni approssimate.

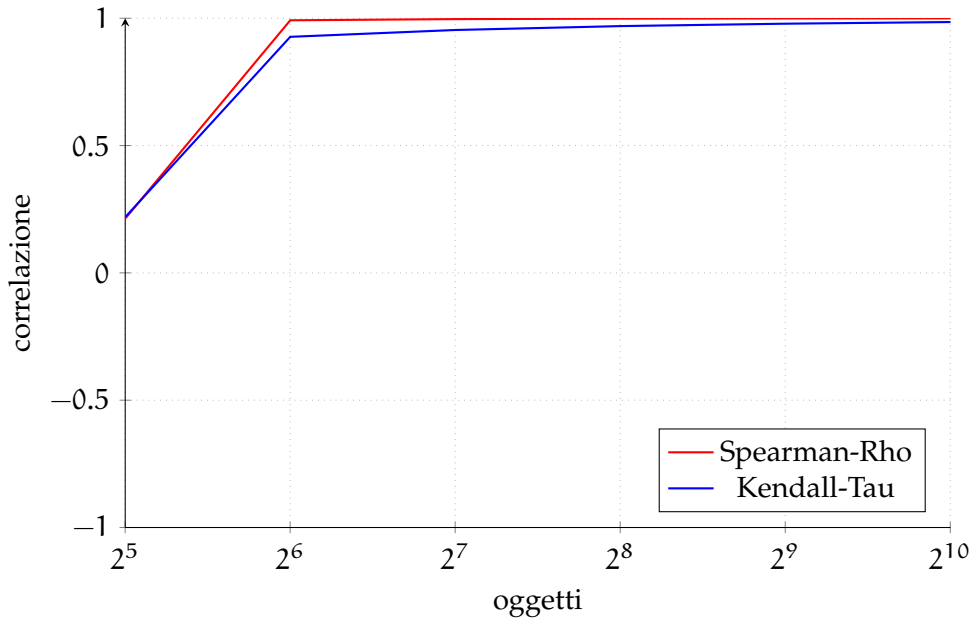

Figura 3.5: Correlazione tra Accuracy e F1, per un numero di utenti pari a 32, all'aumentare del numero di oggetti, per 10000 soluzioni approssimate.

Capitolo 4

Conclusioni

Si è analizzato il problema di valutare un sistema di raccomandazione con dati "unari", ovvero in cui la cifra 1 nella matrice di gradimento rappresenta l'aver apprezzato (o acquistato) un oggetto, ma lo 0 può significare che tale contenuto non è piaciuto o che non se ne conosceva l'esistenza.

Se fosse disponibile l'informazione "è stato guardato ma non è stato acquistato" la situazione sarebbe assai diversa, poichè i valori possibili per ogni entry della matrice sarebbero 3.

Il caso unario è più vago, poichè gli 0 possono significare due cose completamente diverse: in un caso l'articolo non è stato gradito dall'utente, nell'altro tale utente potrebbe esserne interessato.

Sul piano computazionale, un approccio che utilizzi la fattorizzazione non negativa (NMF) oppure il latent semantic indexing fallisce in prima istanza perchè fornisce la soluzione banale di rango 1, ovvero la matrice $B = 1$, che è la più semplice rappresentazione di rango basso dei dati contenuti in A. Tale matrice B è tuttavia priva di ogni significato.

Sul piano algoritmico si richiede quindi di applicare una pesante regolarizzazione dei risultati, al fine di ottenere un numero di 1 statisticamente compatibile con la quantità misurata nella realtà. Un approccio come quello descritto, che non va nella direzione di *fittare* i dati, corre il rischio di sporcare i risultati, rendendoli più aleatori.

In questo lavoro abbiamo analizzato le misure più usate in letteratura per stimare come un eventuale algoritmo possa dare buone o cattive predizioni.

Sono state valutate misure molto classiche, sia nella arcinota categoria del *rating*, come RMSE e errore assoluto, che in quella delle *misure di classificazione*. Appurato che recall e precision vanno in direzioni opposte si preferisce calcolarne una media sofisticata quale F1. Anche Accuracy è un membro interessante di questa categoria e quantifica esattamente la percentuale di elementi correttamente suggeriti e non suggeriti.

Considerazioni diametralmente opposte valgono per le *metriche di ranking*, che risentono della natura intrinsecamente non ordinabile dei dati unari. Al contrario dei dati numerici reali, che si possono ordinare facilmente, per i dati unari il ranking col-

lassa in una ripartizione in due insiemi, all'interno dei quali non c'è un ordinamento naturale.

In information retrieval una misura largamente utilizzata è l'indice di Gini, che vede le sue origini all'inizio del secolo scorso ed era usato per quantificare la sperequazione nel benessere della popolazione. Questa metrica è dimostrato essere di fatto equivalente ad AUC e di risentire pertanto dello stesso problema dell'ordinamento.

All'interno di questo contesto si è cercato di riscrivere e di implementare le misure più note di questa categoria (MAP, AUC e NDCG) ed analizzarne la correlazione fra tutte le possibili coppie.

Poichè la sperimentazione sugli algoritmi non fa parte di questa tesi sono stati usati come valori di partenza matrici A scelte in modo statico e matrici approssimate a valori random in $[0, 1]$, per studiare come tali matrici venissero valutate da queste misure.

I risultati sono ottenuti impiegando le misure di correlazione Spearman ρ e Kendall τ. Si è osservato che le due metriche usate per valutare la correlazione vanno nella stessa direzione e che le prime quattro misure (di rating e di classificazione), ovvero quelle più semplici correlano bene tra loro, mentre le misure di ranking hanno una bassa correlazione sia tra loro che con le altre.

Si ritiene che la variazione di tali risultati sperimentali con l'utilizzo di dati algoritmici sia minima, ma non è evidente questo fatto.

A seguito dell'analisi svolta, si conclude che le misure di ranking non sono facilmente applicabili ad un problema così mal posto. Le metriche di classificazione e di rating invece possono essere utilizzate in modo interscambiabile.

L'osservazione sull'inconsistenza delle metriche di rating trova sostegno nei dati ottenuti dalle sperimentazioni nel caso in cui A sia a valori nell'intervallo $[0, 5]$ dove il valore 0 rappresenta che un articolo non è stato visualizzato, il valore 1 che il giudizio in proposito è negativo, mentre ad un giudizio positivo è assegnato il valore 5. In [4] infatti i risultati sperimentali forniscono un'ottima correlazione tra le metriche di ranking, come si può vedere dalla tabella 4.1.

	CMAE	**CRMSE**	**F_1**	**ACC**	**MAP**	**AUC**	**NDCG**
CMAE	100	90.2	93.4	98.1	86.5	88.6	80.2
CRMSE	90.2	100	75.3	83.8	91.7	92.9	88.
F_1	93.4	75.3	100	96.5	69.6	72.7	61.8
ACC	98.1	83.8	96.5	100	79.9	82.2	72.4
MAP	86.5	91.7	69.6	79.9	100	99.3	98.2
AUC	88.6	92.9	72.7	82.2	99.3	100	96.8
NDCG	80.2	88.	61.8	72.4	98.2	96.8	100

Tabella 4.1: Tabella che rappresenta la correlazione tra le metriche presentate nel Capitolo 1 nel caso di matrici a valori nell'intervallo $[0, 5]$ [4]

Bibliografia

[1] Michael W Berry. Large-scale sparse singular value computations. *The International Journal of Supercomputing Applications*, 6(1):13–49, 1992.

[2] John S Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[3] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[4] Gianna M Del Corso and Francesco Romani. Matrix factorizations and measures comparison for recommender systems. unpublished work, 2018.

[5] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[6] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 5(4):19, 2016.

[7] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.

[8] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434. ACM, 2008.

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[10] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[11] Benjamin M Marlin and Richard S Zemel. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*, pages 5–12. ACM, 2009.

[12] Benjamin M Marlin, Richard S Zemel, Sam T Roweis, and Malcolm Slaney. Recommender systems, missing data and statistical model estimation. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 2686, 2011.

[13] Harald Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 713–722. ACM, 2010.

[14] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. A theoretical analysis of NDCG type ranking measures. *CoRR*, abs/1304.6480, 2013.

[15] Kangning Wei, Jinghua Huang, and Shaohong Fu. A survey of e-commerce recommender systems. In *Service systems and service management, 2007 international conference on*, pages 1–5. IEEE, 2007.

Appendice A

Latent semantic indexing

In questo allegato si espone, in termini formali, che cosa sia un *latent factor*. Si è scelto di spiegare questo concetto applicato a documenti di testo, poichè lo si ritiene più chiaro.

L'obiettivo è quello di assegnare un punteggio ad un insieme di documenti, sulla base della loro pertinenza rispetto ad un gruppo di parole, detto *query* (o richiesta).

Più in dettaglio, nella prima parte vengono gettate le fondamenta per introdurre lo *sketch*, che rappresenta sinteticamente le informazioni sui documenti e le parole. Nella seconda parte, in base a questo *sketch*, viene definita la pertinenza di un documento in relazione ad una *query*.

A.1 I dati

SIA D un insieme di documenti di cardinalità d.

SIA P l'insieme di tutte le parole che compongono i d documenti, con $|P| = p$.

SIA Q l'insieme delle parole che formano la *query*, di cardinalità l.

SIA $M \in M(p, d, \mathbb{N})$ la matrice delle occorrenze dell'insieme P nell'insieme D, in particolare $(M)_{ij}$ rappresenta quante volte il termine p_i occorre nel documento d_j.

SIA $B = M^t M \in S(p, \mathbb{N})$ la matrice di elementi b_{ij}, che rappresentano il numero di coppie di parole uguali tra il documento d_i ed il documento d_j.

SIA $C = MM^T \in S(d, \mathbb{N})$ la matrice di elementi c_{ij}, che rappresentano il numero di coppie di parole (p_i, p_j) in ogni documento.

A.2 Il modello

Con i dati definiti come sopra è possibile decomporre M mediante la fattorizzazione SVD, ottenendo tre matrici:

MATRICE SINISTRA DEI VETTORI SINGOLARI $S \in M(p, r, \mathbb{R})$

MATRICE DESTRA DEI VETTORI SINGOLARI $U \in M(d, r, \mathbb{R})$

MATRICE DIAGONALE DEI VALORI SINGOLARI $\Sigma \in D(r, \mathbb{R})$

tali che $M = S\Sigma U^T$. La matrice Σ ha, sulla diagonale, elementi decrescenti; di conseguenza gli ultimi elementi possono essere trascurati. Si può decidere di ridurre la matrice Σ ad una matrice in $M(k, \mathbb{R})$, ottenendo Σ_k, S_k e U_k, per calcoli più veloci e per ridurre il rumore dovuto a dati non significativi.

$M_k = S_k \Sigma_k {U_k}^t$ è di dimensione $p \times d$, come M, e la approssima.

A.3 I concetti nascosti

In modo un po' informale è possibile definire che cos'è un "concetto nascosto": come la diagonalizzazione di una matrice descrive, tramite gli autovettori, delle direzioni privilegiate dalle quali guardare la trasformazione, così gli autovettori in S ed in U rappresentano l'informazione sui documenti e sui dati in modo più intuitivo.

Lo *sketch* di questo algoritmo risiede nell'interpretazione di $S_k\Sigma_k$ e $\Sigma_k U_k^T$ come rappresentazione essenziale di parole e di documenti in termini di combinazione dei concetti.

Di conseguenza la *query* è un concetto modellato come $q = \frac{\sum_{i=1}^{p} (S_k \Sigma_k)_i}{p} \in \mathbb{R}^k$.

Concludendo, la pertinenza di un documento d_i è espressa dalla distanza del coseno tra i due vettori d_i e q, ossia $\frac{d_i q}{|d_i||q|}$.

Appendice B

Dati per le sperimentazioni

Tra i numerosi dataset disponibili gratuitamente in rete, solo alcuni sono composti da dati unari. Si riportano di seguito i dataset analizzati, atti a facilitare la ricerca futura sull'argomento.

CARRELLI UNICOOP FIRENZE. Contiene informazioni sugli acquisti effettuati dai clienti Unicoop Firenze. Per ogni utente sono presenti le distanze del suo indirizzo di residenza da ciascuno dei supermercati presi in considerazione e il numero di oggetti di ciascun tipo acquistati in ogni supermarket. Inoltre è anche presente il costo di un oggetto in ogni supermercato. Sono presenti 60 366 utenti, 5 supermercati, 4 567 prodotti e 24 638 725 acquisti effettuati (19 960 785 considerando equivalenti gli acquisti dello stesso prodotto in supermercati diversi).

VISUALIZZAZIONI, INSERIMENTI CARRELLO, ACQUISTI in un *e-commerce*: contiene informazioni sulle operazioni effettuate da vari utenti su vari oggetti su un sito di acquisti. Ogni oggetto può essere visto, messo nel carrello o comprato. Contiene informazioni su 1 407 581 utenti e 235 062 oggetti. Il numero di visualizzazioni è 2 664 312, il numero di aggiunte a carrello 69 332 e il numero di acquisti a 22 457. Inoltre per ogni evento è fornito il timestamp in cui è avvenuto.

NETFLIX discretizzato su $\mathcal{V} = \{0, 1\}$. Il dataset originale contiene dati su 17 770 film e 480 189 utenti, nonchè 100 480 507 valutazioni di film (con un numero intero da 1 a 5) da parte di questi utenti. Sono anche disponibili le date dell'assegnazione del voto, la data di uscita del film e i loro titoli. Per rendere unario il dataset basta considerare come 1 tutti i rating di almeno 3 stelle e come 0 tutti i rating inferiori o assenti.

CLICK SU ARTICOLI di una biblioteca digitale; questo dataset è il più minimale per trattare il problema unario, infatti si hanno solo i click.