

Metodo k-fold cross-validation

- Finoto k si divide il dataset in $D_1, D_2 \dots D_k$ di items di m .
- Alla i -esima iterazione con i tra 1 e k si usi D_i come test e il resto come TR.
- Leave-one-out: Variante dove le ripartizioni sono di triple. Anche questo metodo potrebbe essere implementato giusto per calcolo delle medie.

Sistemi di Raccomandazione

Sistemi che cercano di prevedere preferenze di un utente sulla base delle preferenze espresse in precedenza. Esempio: suggerire articoli in base agli articoli letti in precedenza.

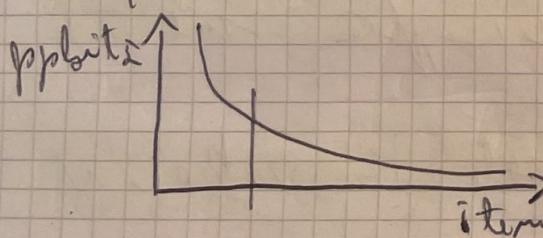
Esistono 2 tipi principali di raccomandazione:

- Sistemi Content-based: effettuano raggiamenti basati sugli items.
- Sistemi collaborativi filtering: suggeriscono sullo stesso basis similità tra utenti e items. Suggerisco ad un utente x items preferiti da altri utenti simili a x .

In sostanza i content-based sono basati sugli items mentre i secondi sono basati su similità tra utenti.

Fenomeno Long Tail

Ordino item dal più popolare a quello gli item in funzione della popolarità.



I sistemi che non usano la raccomandazione non considerano la lunga coda.

Esvolgono:

- Per funzionare bene chi intera la raccomandazione deve avere avuto almeno due dati.
- Più dati abbiamo più è sicura la prefezione.
- I criteri di raccomandazione possono essere usati anche per propagarla
- Il criterio di raccomandazione rientra in un insieme di preferenze conosciute espresse da Utenti per degli items.
- L'utente rappresentato dalla righe: Utenti
- Colonne: items

$M[i, j]$ potrebbe essere booleano se all'utente i piace l'item j oppure un valore numerico (0.1-60 123) che indica il grado di preferenza di i per j.

La matrice potrebbe avere entry nulle che indicano che non si conosce la preferenza di i per j.

L'obiettivo è prevedere i valori non scritti di M.

Costruire la matrice di utilità

- **Approccio esplicito:** chiedere valutazione / recensione all'utente
- **Approccio implicito:** apprendere la valutazione dal comportamento dell'utente

Sistemi Content-based

Idea: suggerire all'utente U oggetti simili a quelli già acquistati da U.

Il sistema è basato sulle proprietà dell'oggetto.

A G ogni ~~oggi~~ item è associato un vettore di elementi. Ogni elemento è una proprietà.

Oggetti simili hanno vettori simili.

Bisogna costruire il profilo anche di U per l'item I.

A U per l'item I si associa un vettore di n elementi.

Questi elementi rappresentano il grado medio di preferenza espresso dall'utente per gli item contenenti quella proprietà.

Lo si calcola tra il profilo di U e quello di I rappresentato da preferenze predette su U su I .

Schemi generali

- 1) tra le proprietà conosciute per gli oggetti che U ha già voluto scegli p_1, p_2, \dots, p_k proprietà.
- 2) Per ogni I voluto da U costruire il suo profilo ovvero un vettore di k elementi dove $c_j = 1$ se l'item ha la proprietà p_j e 0 se no.
- 3) Costruire profilo di U cioè vettore di k elementi dove u_j è il grado medio di preferenza di U per gli item voluti da U contenenti la proprietà p_j .
- 4) Costruire profilo di item I' non voluto da U .
- 5) Calcola similitudine tra profili di U e di I' per predire il grado di preferenza di U per I' .

Profili di un item

descrivono quali proprietà contiene un item.

Le proprietà rappresentate

category, tag e parole chiave da associare a item

Ne documenti si fa text mining

$$TF_{ij} = \frac{f_{ij}}{\max_k p_k} \quad IDF_i = \log \frac{N}{n_i} \quad TF \cdot IDF_{ij} = TF_{ij} \times IDF_i$$

N : documenti, n_i : doc i che hanno parola i.
 f_{ij} : frequenza parola i nel doc j.

Profilo Utente

Portavoce delle valutazioni già effettuate su altri item
è perciò "unire" le valutazioni che riguardano item
con lo stesso profilo.

Funzione aggregazione media: media valutazioni

Esempio: ho valutato Batman 2, Star Wars 10 e
Capitan N. Il primo non è valutato Comeby, gli altri
sono. Profilo utente per Comeby $\Rightarrow \frac{10+8}{2} = 9$
Faccio così per ogni genere dei film
e ottengo un vettore (profilo utente)

Similitudine tra profili

Voglio la misura del simile

Dati 2 vettori di k elementi: il profilo dell'utente
e il profilo dell'item.

$$\text{cosim}(\vec{U}, \vec{V}) = \frac{\vec{U} \cdot \vec{V}}{\|\vec{U}\| \cdot \|\vec{V}\|} = \frac{\sum_{i=1}^k u_i v_i}{\sqrt{\sum_{i=1}^k u_i^2} \sqrt{\sum_{i=1}^k v_i^2}}$$

Tieni conto della direzione

dei vettori e non solo della distanza euclidea

I vettori variano da -1 (antiparalleli) a 1 (paralleli)

Ottieni i vettori da -1 a 1 (nel nostro caso da
0 a 1 nei film con voti da 0 a 10) e moltiplica
per il lunghezza aggiungendo $\text{sim} - 0$.

Esempio: voti da 2 a 10: ottenuta 0,6 per la
lunghezza dell'utente $\Rightarrow (0,6 \cdot 8) + 2$

Punti di debolezza:
• Se ci sono profili
nuove?

• Cosa significa item?

• Entrà nuovo utente con
profilo riconosciuto (la sua lunghezza
nella matrice di utilità è nulla)

Caso: facile, non richiede
interazione con altri
utenti.

Sistema Collaborativo Filtering

Idee: di oggetti da suggerire a U sono quelli
voluti meglio da utenti simili a U

Il rating migliore a U item giochi a Utenti
simili a U.

E' più incentrato sul comportamento utenti

Differenze rispetto al content-based

• Profil item → colonne matrice utilità

• Profil utenti → righe matrice utilità

Schemi generali (base in similarità utenti)

1) Individua N utenti simili a U che hanno
volutato I

2) Calcola media rating dei rating degli altri degli
N utenti su I. I valori sono regni simili
di similitudine tra i concetti degli N utenti su I

3) Il valore è il rating predetto per U su I

Similarità tra utenti

Il profilo dell'utente è rappresentato dalla sua
riga nella matrice di utilità.

N.B.: voto mancante \Rightarrow 0

Viamo la distanza del coseno per le distanze
tra utenti.

Dobbiamo però fare attenzione alla differenza tra
0 per mancanza dati e 0 per valutazione bassa

Dobbiamo centrare i valori rispetto a 0.

Rating bassi \Rightarrow valori negativi

Rating alti \Rightarrow valori positivi

E così fare sottraiendo a ogni rating di rating
conosciuto la media dei rating assegnati dall'utente
a quei item. (dove c'è conoscenza $\Rightarrow -m$) (in qualche
caso vorremo prendere il rating per un item da
un utente).

- Prendo N elementi/profilo più vicini (distanza minima)
- Calcolo la media resta dei rating nulli; tra i N profili (NB: il zero dipende dalla distanza)
- Il valore calcolato è il valore predefinito.

Schemi alternativi (base a similarità item)

- 1) Individua N item più vicini all'item I su cui l'utente ha effettuato il rating
- 2) Calcola media resta dei rating dati dall'utente sugli N item.
I due sono dati in base alla similarità tra gli N item ed I
- 3) Il valore ottenuto è il rating predefinito per I .
Sarebbe uno schema generale ma qui si parla della similarità tra item.

Profilo item = colonne matrice utility

La similarità si potrebbe calcolare anche con la dist del coseno.

Confronto

Schemi basati su similarità item:

- più informativo e affidabile (ma più lento)

Schemi basati su similarità utenti:

- più efficiente se vogliamo prendere i rating dell'utente U

Pregi del collaborazione filtering

- lavora con tutti gli item anche se non proprieti diretti
- Non serve una relazione di proprietà/pertinenza degli item

Difetti:

- impossibile creare predizione per nuovi utenti
visto lo schema basato su relazioni tra item
- impossibile creare predizione riguardante nuovi item con lo schema che non hanno degli utenti

Nella pratica

I sistemi di raccomandazione sono degli ibridi tra content-based e collaborazione filtering.

Singular Value Decomposition SVD

I sistemi di classificazione potrebbero avere un'attuale attivita di utilizzo.

Dobbiamo far fronte all'efficienza e al problema della dimensionalità

Dobbiamo abbattere tecniche d'dimensionalità
reduzione che permette di lavorare con vettori e matrici più piccole

Una tecnica utile per ridurre la dimensionalità è il clustering

Applico gli item id clustering per assegnare le colonne degli item una o più colonne che hanno i valori medi (quanti il cluster)

Praticamente utile ma 2 volte inefficiente se ci sono tanti dati

Questo giudizio richiede di fare clustering su tanti dati. Esempio su slide

Decomposizione di matrici

Potremmo trovare un insieme "piccoli" di feature per item o utenti discriminanti.

Queste feature sono rappresentative di intere categorie o gruppi di feature

Dato l'insieme di feature più importanti di altre ci viene un'idea: usare tecniche di decomposizione della matrice che riportano a 2 matrici esprimendo come prodotto di 2 matrici

$$m \times n \Rightarrow m \times r \quad r \times n$$

In altri algoritmi $m \times n \Rightarrow m \times r \quad r \times n \quad r \times n$

Fra le tecniche di decomposizione c'è la SVD

SVD

Dato M $m \times n$ si esprimiamo nel prodotto di 3 matrici

$$M = U \Sigma V^T$$

• U : matrice ortogonale $m \times r$ cioè $U U^T = I$ di dimensione $m \times r$

• Σ : matrice diagonale $r \times r$ dove non ci sono numeri negativi

• V : matrice ortogonale $n \times r$ ($V^T = r \times n$)

La decomposizione si chiama scomposizione

Gli elementi di Σ sono tutti valori singolari di M . Il numero di valori non nulli di Σ è il rango di M .

colonne di U e V sono dette vettori singolari
sinistri e destri di M

- vettori singolari sinistri di M : autovettori di MM^T
 - vettori singolari destri di M : autovettori di M^TM
- I valori singolari di M sono le radici quadrate degli autovalori di MM^T e M^TM

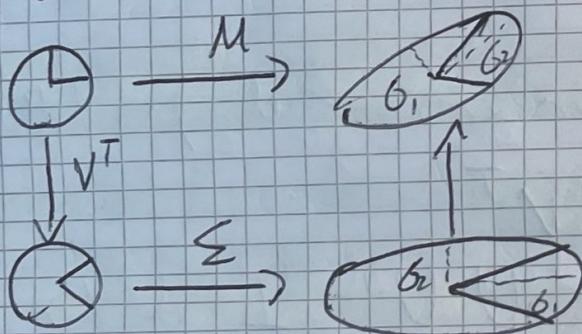
Per trovare la decomposizione SVD abbiamo
trovare autovettori e autovalori di MM^T e M^TM

Interpretazione geometrica di SVD

$$M: 2 \times 2$$

La SVD ruota e trasforma il cerchio in un ellisse dove i semiasse hanno lunghezze pari ai valori singoli non nulli di M .

Cerchio: fascio unitario in \mathbb{R}^2 dove i "frazzi"
sono i due vettori unitari canoni.



$$M = U \cdot \Sigma \cdot V^T$$

Si può applicare SVD sulla matrice N
UNIFI per avere 3 matrici più piccole
dove i conti da fare saranno più facili
e veloci

In pratica avendo n utenti e n item ritengo
mappando i dati in un nuovo spazio con
 m , n e r dove r sono le categorie.

$$r = \text{rang}(M) = |\text{categorie}|$$

Dato M con m utenti e n item la
corrispondente matrice è $U \in VT$ dove:

- U è $m \times r$ (m utenti, r t.categorie)
- Σ è $r \times r$
- V è $n \times r$ (n item e r categorie)

Con SVD primo calcolo delle predizioni di un
utente per tutti gli item con prodotto tra U e V .

- 1) Utente è X vettore di m ratings per gli item.
(Rating non scritto $\equiv 0$)
- 2) Si moltiplica X per V (risparmia i rating
sui spazi delle categorie)
 $Y = X V \leftarrow Y$ è un vettore con i rating per le
categorie.
- 3) Si moltiplica Y per V^T (risparmia i rating
sui spazi delle categorie sulle spazio originale)
- 4) $R = Y V^T$ è un vettore che contiene le
predizioni dei rating di U negli item

Quante sono le categorie? (r) R è un vettore
di dimensione r che ha i
rating per gli item

Esempio su ride

Valutazione binaria

Approccio binario = una valutazione di raccomandazione in
gradi vedere come un classificatore o
preditore.

Possiamo valutare l'offerta b facendo i valori binari
di rating.

Supponiamo la matrice di utilità sia binaria
ovvero like / not like (oppure preferisco
non direne nulla) mettere il rating in alto/basso.

Possiamo a questo punto usare le misure della
matrice di confusione vista per la classificazione.

True positive: L'utile che il rating ha predetti.

True negative: Non utile che il rating ha negato.

Possiamo valutare l'affidabilità del rating
ella curva ROC ad esempio (o comunque altre
curve con il medesimo scopo).

Approccio grande (non binario)

Ciò significa che ci sono rating di grande
tasso in media il ~~predetto~~ rating di
raccomandazione.

RMSE = Root Mean Square Error

$$RMSE(\hat{R}, R) = \sqrt{\frac{\sum_{i=1}^{|R|} (\hat{R}_i - R_i)^2}{|R|}}$$

\hat{R} = vettore rating predetti.

R = vettore rating veri \Rightarrow Stessa entità.

Questo rating di valutazione è buono per
valutare rating di raccomandazione non binari
coi non binari solo sul like/not like.

Un esempio è un rating dove ogni item
ha un rating (da parte di un utente) da
0 a 5