

# Programma Paper 13

Il programma sarà scritto nel linguaggio Python 2.7.12 e sarà composto dai seguenti file:

## 1. *Predictors.py*

File contenente la classe Predictors che sarà suddivisa nei seguenti moduli:

- ❑ IKNN: Item-based K-nearest neighbors

data la similarità tra gli items  $i$  e  $j$ :

$$sim_{I,O}(i,j) = \frac{s_{I,O}(i,j)}{(s_I(i)+\lambda)^\alpha * (s_I(j)+\lambda)^{1-\alpha}}$$

dove:

$I$                       Insieme degli Input delle tipologie di eventi  
 $O$                       Insieme degli Output delle tipologie di eventi

$s_{I,O}(i,j)$             Calcolo delle co-occorrenze basate tra gli input e gli output tra gli item  $i$  e  $j$ , viene definita nel seguente modo:

$$s_{I,O}(i,j) = \sum_u w_I(u,i) * w_O(u,j)$$

dove:

$I$                       Insieme degli Input delle  
tipologie di eventi  
 $O$                       Insieme degli Output delle  
tipologie di eventi  
 $w_I(u,i)$  peso del più recente evento  $e$   
avvenuto sull'item  $i$  dall'utente  $u$   
, assumendo che:

$$I(e) \in I$$

$I(e)$       tipo di evento  $e$

Nel caso  $I(e) \notin I$ , allora sarà:

$$w_I(u,i) = 0$$

$w_O(u,j)$             peso del più recente evento  $e$   
avvenuto sull'item  $j$  dall'utente  $u$ ,  
assumendo che:

$$O(e) \in O$$

$O(e)$       tipo di evento  $e$

Nel caso  $O(e) \notin O$ , allora sarà:

$$w_O(u,j) = 0$$

$\lambda$                       Coefficiente di regolarizzazione

$\alpha$                       Esponente di normalizzazione

(usato allo scopo di penalizzare il fattore di popolarità)

$sim_{I,O}(i,i) = 0$  come scelta di design.

Il predittore dell'item  $i$  per l'utente  $u$  è definito come:

$$p(u, i) = \left( \sum_{e \in E(u)} w(e) \right)^{-1} * \sum_{e \in E(u)} w(e) * \text{sim}_{I,O}(i(e), i)$$

dove:

$w(e)$  peso dell'evento  $e$ , viene definito nel seguente modo:

$$w(e) = w_y(e)(1 - \delta^{t_{max} - t(e)})$$

dove:

$w_y(e)$  peso costante assegnato ad un tipo di evento

$\delta$  valore indicante il decadimento temporale

$t_{max}$  ultimo orario segnato nel training set

$t(e)$  tempo nel quale è stato segnalato l'evento  $e$

$E(u)$  insieme degli eventi dell'utente  $u$

$i(e)$  item dell'evento  $e$

#### ❑ RCTR: Recalling recommendations

Predittore basato sul concetto che esiste una significativa probabilità di effettuare una raccomandazione su un item di cui si è già fatta una raccomandazione nelle settimane precedenti.

Il predittore dell'item  $i$  per l'utente  $u$  è definito come:

$$p(u, i) = \sum_{e \in E(u)} \frac{s_C(i(e))}{s_R(i(e))} * 1_A(y(e) = 5)$$

dove:

$C$  Clicks

$R$  Impressions

$1_A$  Funzione indicatrice

$i(e)$  item dell'evento  $e$

$y(e)$  tipo dell'evento  $e$

$E(u)$  insieme degli eventi dell'utente  $u$

❑ AS: Already seen items

Predittore basato sul concetto che esiste una significativa probabilità di ritornare su un item visto in precedenza.

Il predittore dell'item  $i$  per l'utente  $u$  è definito come:

$$p(u, i) = \sum_{e \in E(u)} w(e) * 1_A(y(e) \in I)$$

dove:

- $w(e)$  peso dell'evento  $e$
- $1_A$  Funzione indicatrice
- $I$  Insieme degli Input delle tipologie di eventi
- $y(e)$  tipo dell'evento  $e$
- $E(u)$  insieme degli eventi dell'utente  $u$

❑ UPOP: User metadata-based popularity

Predittore basato sul concetto che moltissimi dei target users sono nuovi.

Per essi non risulta possibile usare le interaction ed impressions.

La popolarità di un item  $i$  per un gruppo di utenti  $g$  viene calcolata come:

$$pop_g(i) = \frac{1}{|U(g)|} * \sum_{u \in U(g)} \sum_{e \in E(u)} w(e) * 1_A(i = i(e))$$

dove:

- $w(e)$  peso dell'evento  $e$
- $U(g)$  Insieme di utenti del gruppo  $g$
- $1_A$  Funzione indicatrice
- $E(u)$  insieme degli eventi dell'utente  $u$

Il predittore dell'item  $i$  per l'utente  $u$  è definito come:

$$p(u, i) = \frac{1}{|G(u)|} * \sum_{g \in G(u)} \frac{pop_g(i)}{pop_g(i) + \lambda}$$

dove:

- $G(u)$  gruppo a cui appartiene l'utente  $u$
- $\lambda$  Coefficiente di regolarizzazione

❑ MS: Meta cosine similarity

In tale predittore la similarità degli item  $i$  e  $j$  viene calcolata come la similarità coseno dei loro vettori ( $sim(i,j)$ )

Il predittore delle preferenze dell'utente  $u$  per gli items  $i$  è definito come:

$$p(u, i) = \left( \sum_{e \in E(u)} \right)^{-1} * \sum_{e \in E(u)} w(e) * sim(i(e), i)$$

dove:

$w(e)$  peso dell'evento  $e$

$E(u)$  insieme degli eventi dell'utente  $u$

❑ AP: Age-based popularity change

Tale predittore è basato sul fatto che è presente un abbassamento della popolarità tra gli users con un'età tra i 25-30 anni e tra gli items con tra 0 55-60 giorni.

Il predittore è definito come segue:

$$p(u, i, t) = \frac{\sum_{k=1}^7 (8-k) * pop_a(age(i,t)+k)}{28 * pop_a(age(i,t))}$$

dove:

$pop_a(d)$  popolarità presunta di un item  $a$  con età  $d$

$age(i, t)$  età di un item  $i$  al tempo  $t$

di default viene considerata  $p(u, i) = p(u, i, t_{max})$

## 2. **Data\_Read.py**

File contenente la classe Data\_Read, che si occuperà della lettura del dataset e della creazione della struttura dati.

## 3. **Main.py**

File contenente il main del programma, si interfacerà rispettivamente con i file:

❑ **Data\_Read.py**

Allo scopo della creazione della base di dati

❑ **Predictors.py**

Allo scopo del calcolo dei vari predittori

All'interno del main saranno effettuate le varie ottimizzazioni ed infine saranno combinati i risultati dei vari predittori.