

Combinare sistemi di reputazione e di
raccomandazione per ottenere consigli migliori
online

Alberto Bezzon, Tommaso Carraro, Matteo Ceradini

20 giugno 2018

Indice

1	Introduzione	5
1.1	Sistemi di raccomandazione	5
1.2	Sistemi di reputazione	5
1.3	Combinazione tra sistemi di raccomandazione e sistemi di reputazione	5
2	Soluzione di Josang	7
2.1	Introduzione	7
2.2	Opinioni soggettive	7
2.2.1	Formazione e rappresentazione	7
2.2.2	Tipologie di opinioni soggettive	8
2.2.3	Conversione di opinioni multinomiali in opinioni binomiali	8
2.3	Determinazione delle opinioni	9
2.3.1	Opinioni derivate dai sistemi di reputazione	9
2.3.2	Opinioni derivate dai sistemi di raccomandazione	10
2.4	Combinazione dei valori di raccomandazione e reputazione	11
2.4.1	Caratteristiche dell'operatore CasMin	12
2.4.2	Operatore CasMin con opinioni multinomiali	12
2.4.2.1	Pseudocodice	13
2.4.2.2	Esempio	14
2.4.3	Operatore CasMin con opinioni binomiali	15
2.4.3.1	Pseudocodice	15
2.4.3.2	Esempio	16
2.5	Conclusioni	16
3	Una nostra alternativa	17
3.1	Introduzione	17
3.2	Descrizione	17
3.3	Sviluppo	18
3.4	Esempio	18
3.5	Conclusioni	19
4	La nostra implementazione	20
4.1	Script in Python	20
4.1.1	Reperimento del codice	20
4.1.2	Dataset di input	20
4.1.3	Logica applicativa	20

	4.1.4	Dataset di output	21
4.2		Applicazione web	21
	4.2.1	Reperimento dell'applicazione web	21
	4.2.2	Tabelle sorgenti	21
	4.2.3	Risultati	21
	4.2.3.1	Combinazione finale	22
	4.2.3.2	Algoritmo di CasMin	22
	4.2.3.3	Vincoli soft	22
	4.2.4	Ricerca dettagliata	22
5		Bibliografia	23

Elenco delle figure

1	Esempio con vincoli soft	19
---	------------------------------------	----

Elenco delle tabelle

1	Esempio 1 - Risultato CasMin con opinioni multinomiali	14
2	Esempio 2 - Risultato CasMin con opinioni binomiali	16
3	Dati risultati da rep. and rec. system	18

1 Introduzione

1.1 Sistemi di raccomandazione

Su Internet la quantità di informazioni è travolgente e continuamente in aumento, questo ha reso impossibile l'accesso tempestivo a elementi e informazioni di interesse sulla rete. I sistemi di recupero delle informazioni (ad esempio i motori di ricerca) hanno parzialmente risolto il problema ma priorità e personalizzazione (mappare i contenuti disponibili per gli interessi e le preferenze dell'utente) di informazioni erano assenti. Per tale motivo sono stati introdotti i sistemi di raccomandazione.

I *Sistemi di Raccomandazione* sono sistemi di filtraggio di informazioni correlate alle preferenze dell'utente, all'interesse o al comportamento osservato su di esso rispetto alla grande quantità di informazioni. Questi sistemi hanno la capacità di prevedere se un determinato utente preferirebbe un articolo o meno in base al profilo dell'utente.

I sistemi di raccomandazione portano dei vantaggi sia agli utenti, mostrando informazioni di interesse, sia per i fornitori di servizi perché riducono i costi di transazione per la ricerca e la selezione di articoli in ambiente di shopping online. Essi migliorano le entrate in quanto sono mezzi efficaci per vendere più prodotti.

Lo scopo dei sistemi di raccomandazione è principalmente di generare suggerimenti su risorse di cui un utente, a priori non ne è a conoscenza ma probabilmente potrebbe essere interessato ad esse.

1.2 Sistemi di reputazione

I *Sistemi di Reputazione* sono un componente essenziale di varie piattaforme online, come ad esempio siti di e-commerce o sistemi di condivisione di file. Questi sistemi incoraggiano gli utenti a fornire un feedback su esperienze passate. Nascono perché, prima di essi, gli utenti che interagivano con nuovi siti web non avevano alcuna informazione.

Lo scopo dei sistemi di reputazione è quello di fornire consigli su risorse di cui l'utente è interessato e di cui ne è già a conoscenza.

1.3 Combinazione tra sistemi di raccomandazione e sistemi di reputazione

I sistemi di reputazione e raccomandazione sono basati su principi differenti ma hanno entrambi lo stesso scopo: supportare l'utente fornendo delle informazioni per agevolare le sue decisioni. Brevemente, i compiti dei due sistemi sono i seguenti:

- sistemi di raccomandazione: suggeriscono all'utente risorse che non conosce ma alle quali può essere interessato. I valori di raccomandazione sono calcolati basandosi su informazioni fornite da altri utenti per oggetti simili;

- sistemi di reputazione: forniscono alla comunità informazioni riguardanti risorse che l'utente conosce già. I punteggi di reputazione indicano quanto un oggetto è piaciuto a degli utenti.

Per fornire delle raccomandazioni più accurate è possibile integrare i due sistemi. L'integrazione risulta complessa a causa della diversità dei sistemi che hanno forme diverse di feedback. Anche i risultati hanno forme diverse:

- un voto da 1 a 5 per i sistemi di reputazione;
- una tupla (d_1, \dots, d_k) , dove d_i è una diversa caratteristica della risorsa d , per i sistemi di raccomandazione.

Nella sezione §2 viene riassunta una soluzione analizzata e studiata. Nella sezione §3 viene presentata una soluzione che utilizza i vincoli soft.

2 Soluzione di Josang

2.1 Introduzione

La soluzione del Professor Josang e dei suoi colleghi consiste nel fondere i punteggi di reputazione e di raccomandazione tramite un operatore chiamato CasMin (Cascading Minimum Common Belief Fusion). Il problema principale è che i punteggi di reputazione sono potenzialmente diversi dai valori di raccomandazione, ossia sono eterogenei e quindi impossibili da fondere senza prima essere manipolati. Per applicare l'operatore CasMin a tali punteggi bisogna renderli prima di tutto omogenei. Per rendere omogenei i punteggi di reputazione e i valori di raccomandazione, essi devono essere mappati in opinioni soggettive. Dopo essere stati mappati in opinioni soggettive, i punteggi possono essere combinati con l'operatore CasMin che restituirà un valore alto solo se entrambi i punteggi, di reputazione e di raccomandazione rispettivamente, saranno alti. Una risorsa verrà suggerita all'utente solo se è stata raccomandata con alta confidenza e se ha un alto punteggio di reputazione. La raccomandazione diviene più accurata rispetto al solo sistema di raccomandazione che può raccomandare oggetti con bassa reputazione. Infine i consigli distribuiti agli utenti avranno una qualità migliore.

2.2 Opinioni soggettive

In questa sezione viene introdotta la notazione e la formazione delle opinioni soggettive, usate per fondere gusto e fiducia. Successivamente viene descritto un metodo per mappare dei punteggi multinomiali in opinioni binomiali. I punteggi multinomiali sono la forma di feedback più comune restituita dai sistemi di reputazione e dai sistemi di raccomandazione.

2.2.1 Formazione e rappresentazione

Un'opinione soggettiva esprime la credenza (belief) degli stati di uno spazio degli stati chiamato "frame". Uno stato in un frame può essere considerato come una dichiarazione, quindi un frame contiene un insieme di dichiarazioni. Indichiamo con $X = \{x_1, x_2, \dots, x_k\}$ un frame di cardinalità k , dove $x_i (1 \leq i \leq k)$ rappresenta uno specifico stato. Un'opinione distribuisce la massa di credenze sul ridotto powerset del frame, indicato con $R(X)$.

Un'opinione è una funzione composta che consiste in un vettore di belief \mathbf{b} , un parametro di incertezza \mathbf{u} e un vettore di base rate \mathbf{a} , che prende valori nell'intervallo $[0, 1]$ e che soddisfa i seguenti requisiti di additività:

- **Additività belief:**

$$u_X + \sum_{x_i \in R(X)} b_X(x_i) = 1; \quad (1)$$

- **Additività base rate:**

$$\sum_{i=1}^k a_X(x_i) = 1, \text{ dove } x_i \in X. \quad (2)$$

L'opinione di un utente A sul frame X è denotata come $\omega^A_X = (b_X, u_X, a_X)$, dove b_X è un vettore di belief sugli stati di $R(X)$, u_X è la massa di incertezza complementare, e a_X è un vettore di base rate su X , tutti visti dal punto di vista del proprietario delle belief A.

2.2.2 Tipologie di opinioni soggettive

Vengono presentate due diverse tipologie di opinioni soggettive:

- Opinioni **multinomiali**: il vettore di belief b_X viene applicato solamente agli elementi x_i appartenenti a X , e non in $R(X)$;
- Opinioni **binomiali**: vengono applicate ai frame binari e hanno una rappresentazione speciale. Indichiamo con $X = \{x, \bar{x}\}$ un frame binario. Un'opinione binomiale sulla verità dello stato x è la quadrupla ordinata $\omega_X = (b, d, u, a)$, dove:
 - b (belief): è la massa di credenze in supporto che x sia vero;
 - d (disbelief): è la massa di credenze in supporto che x sia falso;
 - u (incertezza): è l'incertezza sulla probabilità di x ;
 - a (base rate): è la probabilità preventiva non informativa di x .

In caso di opinioni binomiali l'equazione (1) è espressa come:

$$b + d + u = 1. \quad (3)$$

2.2.3 Conversione di opinioni multinomiali in opinioni binomiali

Le opinioni multinomiali sono una generalizzazione delle opinioni binomiali e in certi casi può essere necessario proiettare le opinioni multinomiali in opinioni binomiali. Per esempio, un sistema di reputazione dove i punteggi sono dati nella forma 1-5 stelle può rappresentare un punteggio di reputazione come un'opinione multinomiale su un frame con 5 stati, ognuno dei quali rappresenta uno specifico numero di stelle. In questo caso, un punteggio di reputazione rappresentato come un'opinione multinomiale può essere proiettato in un'opinione binomiale nel modo seguente.

Indichiamo con $X = \{x_1, \dots, x_k\}$ un frame dove i k stati rappresentano un livello crescente di punteggio. Indichiamo con $Y = \{y, \bar{y}\}$ un frame binario dove y e \bar{y} indicano un'alta qualità e una bassa qualità di una risorsa,

rispettivamente. Assumiamo che un punteggio di reputazione o un valore di raccomandazione siano espressi tramite l'opinione multinomiale $\omega_X = (b_X, u_X, a_X)$ sul frame X , e che un'opinione binomiale $\omega_Y = (b_Y, d_Y, u_Y, a_Y)$ è richiesta sul frame binario Y . Allora la proiezione dall'opinione multinomiale ω_X su X nella opinione binomiale ω_Y su Y è definita da:

$$\begin{cases} u_Y = u_X \\ b_Y = \sum_{i=1}^k b_{x_i} \frac{(i-1)}{(k-1)} \\ d_Y = 1 - b_Y - u_Y \\ a_Y = \sum_{i=1}^k a_{x_i} \frac{(i-1)}{(k-1)} \end{cases} \quad (4)$$

Il vantaggio di proiettare opinioni multinomiali in opinioni binomiali è la flessibilità di analizzare i punteggi di reputazione e i valori di raccomandazione indipendentemente dalla cardinalità del frame.

2.3 Determinazione delle opinioni

In questa sezione vengono descritte le procedure per derivare opinioni soggettive dai punteggi di reputazione e dai valori di raccomandazione prodotti dai sistemi di reputazione e dai sistemi di raccomandazione, rispettivamente.

2.3.1 Opinioni derivate dai sistemi di reputazione

Un sistema di reputazione generalmente si applica a servizi o beni che possono essere valutati su uno o più aspetti. Nel caso in cui solamente un singolo aspetto può essere valutato, esso rappresenta tipicamente la qualità totale di uno specifico servizio o bene. Ogni aspetto può essere valutato con uno specifico livello di qualità, ad esempio da 1 a 5 stelle. È possibile che per un aspetto possano essere assegnati solo due possibili livelli di qualità, ad esempio “pollice in su” e “pollice in giù”. In quest'ultimo caso i livelli indicano una buona qualità o una cattiva qualità, rispettivamente.

Le opinioni per ogni aspetto possono essere derivate da tali valutazioni. Il frame per ogni aspetto è composto dall'insieme di livelli di punteggio discreti che possono essere assegnati all'aspetto, quindi nel caso in cui per un aspetto si possa assegnare una valutazione da 1 a 5 stelle, il frame corrispondente avrà 5 stati. Indichiamo con X il frame di cardinalità k , con $r(x_i)$ il numero di punteggi assegnati del tipo x_i , e con $\omega_X = (b_X, u, a_X)$ un'opinione multinomiale su X . Più punteggi vengono raccolti e minore sarà l'incertezza. L'opinione ω_X può essere determinata dai punteggi $r(x_i)$ tramite la seguente equazione:

$$\forall x_i \in X \begin{cases} b(x_i) = \frac{r(x_i)}{W + \sum_{i=1}^k r(x_i)} \\ u = \frac{W}{W + \sum_{i=1}^k r(x_i)} \end{cases} \quad (5)$$

Nell'equazione $W = 2$ è il peso precedente non informativo con valore di default imposto dal requisito di avere una funzione di densità di probabilità uniforme su frame binari.

Un punteggio è espresso come uno specifico livello corrispondente ad un singolo stato nel frame. Nel caso in cui dovessero esserci più di due livelli di punteggio, le opinioni derivate saranno multinomiali. Dualmente, nel caso in cui solo due tipi di valutazione possono essere dati, il frame sarà binario e quindi le opinioni derivate saranno binomiali. I punteggi di reputazione espressi come opinioni multinomiali possono essere mappati in opinioni binomiali assumendo che ogni livello di punteggio corrisponda ad un valore compreso nell'intervallo $[0, 1]$.

2.3.2 Opinioni derivate dai sistemi di raccomandazione

Nel nostro progetto i valori di raccomandazione sono generati da un metodo CF (Collaborative filtering) basato sull'utente. Il compito dei metodi CF è di predire la preferenza di una data risorsa per un utente attivo, basandosi sulla storia dell'utente stesso e degli altri partecipanti della comunità. Vengono introdotti i simboli s, v per gli utenti e i, j per gli oggetti. Indichiamo con $r_{v,i}$ il punteggio dato da un utente v all'oggetto i , e con I_v l'insieme degli oggetti che l'utente v ha precedentemente votato. Il punteggio medio dell'utente v è dato dalla seguente formula:

$$\bar{r}_v = \frac{1}{|I_v|} \sum_{i \in I_v} r_{v,i} \quad (6)$$

Indichiamo con $N_{s,j}$ il vicinato di un utente attivo s vincolato dall'aver votato l'oggetto j . Quindi $N_{s,j}$ è l'insieme di utenti che hanno votato gli stessi (o alcuni) oggetti che anche s ha votato e che hanno votato anche lo specifico oggetto j . Generalmente, solo i migliori K utenti più simili a s vengono selezionati per far parte del vicinato. La predizione $p_{s,j}$ per l'utente s sull'oggetto j è data dalla seguente formula:

$$p_{s,j} = \bar{r}_s + k \sum_{v \in N_{s,j}} w(s,v)(r_{v,j} - \bar{r}_v), \quad (7)$$

dove k è un fattore di normalizzazione e $w(s,v)$ rappresenta la somiglianza tra gli utenti s e v . Ci sono diversi metodi per calcolare la somiglianza tra gli utenti ma il più comune è il coefficiente di correlazione di Pearson:

$$w(s,v) = \frac{\sum_{i \in I_{s,v}} (r_{s,i} - \bar{r}_s)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{s,v}} (r_{s,i} - \bar{r}_s)^2 \sum_{i \in I_{s,v}} (r_{v,i} - \bar{r}_v)^2}}, \quad (8)$$

dove $I_{s,v}$ rappresenta l'insieme degli oggetti che entrambi gli utenti s e v hanno votato, e $w(s, v)$ è localizzato nell'intervallo $[-1, 1]$. Un problema per il calcolo della somiglianza è che nel caso in cui non ci sono, o solamente pochi oggetti sono stati votati in comune, la somiglianza calcolata non è affidabile e introduce incertezza rispetto al valore atteso. Questo problema è detto **cold start**. Però, quando le predizioni sono rappresentate in termini di opinioni soggettive, il grado di incertezza può essere esplicitamente espresso. Qui sotto viene descritto un metodo per ricavare opinioni binomiali dalle predizioni fatte da un metodo CF.

La derivazione si basa su due assunzioni intuitive:

- L'incertezza della predizione derivata è una funzione decrescente del numero di punteggi dati dagli utenti simili in $N_{s,j}$. Più utenti ci saranno nel vicinato e più affidabile sarà la predizione risultante;
- Vale l'equazione (3).

Quindi emerge il seguente sistema di equazioni:

$$\begin{cases} u_j^s = \frac{W}{W + \sum_{v \in N_{s,j}} |I_{s,v}|} \\ p_{s,j} = b_j^s + au_j^s \\ 1 = b_j^s + d_j^s + u_j^s \end{cases} \implies \begin{cases} u_j^s = \frac{W}{W + \sum_{v \in N_{s,j}} |I_{s,v}|} \\ b_j^s = p_{s,j} - au_j^s \\ d_j^s = 1 - b_j^s - u_j^s \end{cases} \quad (9)$$

dove $W = 2$ è il precedente peso non informativo. Come detto precedentemente $N_{s,j}$ è il vicinato dell'utente s vincolato dall'aver votato l'oggetto j , e $I_{s,v}$ è l'insieme di oggetti che entrambi gli utenti s e v hanno votato.

Sebbene l'equazione sia ottenuta da un metodo CF basato sull'utente, essa può essere facilmente adattata ai metodi CF basati sull'oggetto:

$$\begin{cases} u_j^s = \frac{W}{W + \sum_{i \in N_{s,j}} |U_{i,j}|} \\ p_{s,j} = b_j^s + au_j^s \\ 1 = b_j^s + d_j^s + u_j^s \end{cases} \implies \begin{cases} u_j^s = \frac{W}{W + \sum_{i \in N_{s,j}} |U_{i,j}|} \\ b_j^s = p_{s,j} - au_j^s \\ d_j^s = 1 - b_j^s - u_j^s \end{cases} \quad (10)$$

dove $N_{s,j}$ è il vicinato dell'oggetto j , ossia l'insieme di oggetti che sono stati votati dagli utenti che hanno votato anche l'oggetto j e dove tutti (o alcuni) sono stati votati dall'utente s . Mentre $U_{i,j}$ è l'insieme degli utenti che hanno votato entrambi gli oggetti i e j . Come prima, solo i migliori K oggetti saranno selezionati come vicinato per le predizioni di punteggio.

2.4 Combinazione dei valori di raccomandazione e reputazione

Dopo aver ottenuto le opinioni soggettive dai sistemi di reputazione e dai sistemi di raccomandazione, esse possono essere combinate. Il gruppo utilizza

l'operatore CasMin introdotto da Josang e i suoi colleghi. Questo operatore ha la capacità di combinare i livelli di punteggio dei due diversi sistemi, se questi sono espressi come opinioni soggettive. In questa sezione vengono presentati i principi e la logica dell'algoritmo CasMin e due versioni dello stesso:

- una versione proposta da Josang che richiede che i punteggi siano espressi come opinioni multinomiali;
- una versione, dal gruppo rivista, basata sulla versione di Josang, che richiede che i punteggi siano espressi come opinioni binomiali.

Infine viene presentato un esempio di utilizzo dell'operatore CasMin, nella realtà di TripAdvisor, con un dataset puramente inventato.

2.4.1 Caratteristiche dell'operatore CasMin

L'operatore CasMin è applicabile quando gli stati nel frame rappresentano livelli ordinati di punteggio. Quando si combinano le masse di credenze del più alto livello sul frame, la migliore massa di credenza tra i due argomenti è ridotta per essere matchata con la massa di credenza del secondo argomento. Questo permette di distribuire omogeneamente le masse di credenza su tale stato (livello di punteggio). La porzione di massa di credenza che è stata precedentemente rimossa dal migliore dei due argomenti viene sommata alla massa di credenza del livello subito inferiore nel frame. L'algoritmo procede in questo modo fino ad arrivare al livello minimo del frame. Infine la massa di credenza in eccesso degli ultimi argomenti può essere matchata con la massa di incertezza dell'altro argomento. In questo modo l'incertezza viene ridotta, mentre la massa di credenza del livello più basso viene incrementata.

Questo operatore gode della **proprietà conservativa**. Questa proprietà è utile a fare in modo che un oggetto venga raccomandato solo se ha un alto valore di raccomandazione e un alto punteggio di reputazione. CasMin fornisce un modello di combinazione conservativo poiché prende il più piccolo punteggio di reputazione e valore di raccomandazione a ogni livello, partendo dal livello più alto, e ad ogni livello fa scendere i valori in eccesso nei livelli inferiori. Quindi un alto valore di CasMin può essere ottenuto solamente se entrambi i sistemi producono punteggi alti. In questo modo, il consiglio prodotto dalla combinazione risultante di CasMin sarà più conservativo rispetto a quello prodotto dai due sistemi isolati.

2.4.2 Operatore CasMin con opinioni multinomiali

Indichiamo con $X = \{x_1, \dots, x_k\}$ un frame ordinato di cardinalità k , dove x_k è il livello di punteggio più alto ottenuto da un sistema di raccomandazione o reputazione. Assumiamo che ci siano due opinioni ω^A_X e ω^B_X sul frame X dove A e B indicano due diversi proprietari di credenze, ossia due argomenti. Le due opinioni possono essere matematicamente combinate usando l'operatore CasMin denotato dalla seguente espressione: $\omega^{(A \downarrow B)}_X = CasMin(\omega^A_X, \omega^B_X)$.

La soluzione proposta da Josang richiede opinioni multinomiali. Prima di tutto agisce sulla credenza del livello più alto x_k e per finire agisce sulla credenza del livello più basso x_1 . Ad ogni livello esegue le operazioni descritte nella sezione precedente. La linea 2 assicura che la credenza dell'argomento A sia sempre migliore della credenza dell'argomento B, facendo un'operazione di **swap** se necessario. Al termine del ciclo la nuova opinione dell'utente A rappresenta il risultato combinato che verrà restituito.

2.4.2.1 Pseudocodice

```

1: for  $i = k$  to 2 do
2:   if  $b^A_X(x_i) \leq b^B_X(x_i)$  then
3:      $Swap(\omega^A_X, \omega^B_X)$ 
4:   end if
5:   if  $u^B_X > (b^A_X(x_i) - b^B_X(x_i))$  then
6:      $u^B_X = u^B_X - (b^A_X(x_i) - b^B_X(x_i))$ 
7:      $b^B_X(x_i) = b^A_X(x_i)$ 
8:      $b_{cascade} = 0$ 
9:   else
10:     $b^B_X(x_i) = b^B_X(x_i) + u^B_X$ 
11:     $u^B_X = 0$ 
12:     $b_{cascade} = b^A_X(x_i) - b^B_X(x_i)$ 
13:     $b^A_X(x_i) = b^B_X(x_i)$ 
14:   end if
15:    $b^A_X(x_{i-1}) = b^A_X(x_{i-1}) + b_{cascade}$ 
16: end for
17:  $CasMin = \omega^A_X$ 

```

Algorithm 1: CasMin con opinioni multinomiali

2.4.2.2 Esempio

Hotel	Rep. Ratings	Multinomial Rep. Score	Binomial Rep. Score	Binomial Rec. Value	CasMin Advice
Hotel 1	$r(x_5) = 50$ $r(x_4) = 10$ $r(x_3) = 10$ $r(x_2) = 0$ $r(x_1) = 5$	$b_{x5} = 0.65$ $b_{x4} = 0.13$ $b_{x3} = 0.13$ $b_{x2} = 0.00$ $b_{x1} = 0.06$ $u_X = 0.03$	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.1$ $d = 0.7$ $u = 0.2$	$b = 0.30$ $d = 0.70$ $u = 0.00$
Hotel 2	$r(x_5) = 5$ $r(x_4) = 0$ $r(x_3) = 10$ $r(x_2) = 10$ $r(x_1) = 50$	$b_{x5} = 0.06$ $b_{x4} = 0.00$ $b_{x3} = 0.13$ $b_{x2} = 0.13$ $b_{x1} = 0.65$ $u_X = 0.03$	$b = 0.16$ $d = 0.81$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.19$ $d = 0.61$ $u = 0.20$
Hotel 3	$r(x_5) = 50$ $r(x_4) = 10$ $r(x_3) = 10$ $r(x_2) = 0$ $r(x_1) = 5$	$b_{x5} = 0.65$ $b_{x4} = 0.13$ $b_{x3} = 0.13$ $b_{x2} = 0.00$ $b_{x1} = 0.06$ $u_X = 0.03$	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.81$ $d = 0.19$ $u = 0.00$

Tabella 1: Esempio 1 - Risultato CasMin con opinioni multinomiali

Nell'esempio illustrato si considera il caso di dover fornire consigli sugli hotel di un sito web. Si assume di avere un sistema di raccomandazione che tracci le preferenze degli utenti, e un sistema di reputazione che permetta agli utenti di valutare gli hotel. Con l'equazione (5) il sistema di reputazione può produrre punteggi espressi come opinioni multinomiali. Con l'equazione (4) le opinioni multinomiali possono essere trasformate in opinioni binomiali. I valori di raccomandazione per ogni hotel e ogni utente sono espressi come opinioni binomiali usando le equazioni (9) (CF basato sull'utente) o (10) (CF basato sull'oggetto). Il sistema di raccomandazione identifica una lista di hotel basandosi sul punteggio dato dall'utente e da altri viaggiatori. Esso può prevedere che all'utente piacerà un hotel perché altri utenti con gusti simili sono stati soddisfatti dell'hotel. Invece, il sistema di reputazione produce punteggi disponibili a tutta la comunità, per ogni hotel. L'operatore CasMin produce risultati conservativi in modo che un hotel debba avere simultaneamente alti valori di raccomandazione e alti punteggi di reputazione.

Nella tabella è mostrata la combinazione di due opinioni multinomiali, una per il sistema di raccomandazione e una per il sistema di reputazione, utilizzando l'algoritmo sopra descritto. Vengono illustrati i risultati di analisi di tre hotel differenti sul sito web. Nei casi dell'hotel 1 e l'hotel 2, dove i valori di

raccomandazione e i punteggi di reputazione sono in conflitto, il consiglio restituito dal CasMin ha un belief basso. L'unico risultato buono è per l'hotel 3, dove entrambi i sistemi hanno prodotto punteggi positivi e la combinazione ha prodotto un buon valore di belief del CasMin.

2.4.3 Operatore CasMin con opinioni binomiali

Il gruppo presenta una versione rivista rispetto a quella di Josang. La versione proposta richiede opinioni binomiali. Prima di utilizzare CasMin bisogna ottenere delle opinioni multinomiali a partire dai punteggi di reputazione prodotti dal sistema di reputazione tramite l'equazione (5). Queste devono essere poi mappate in opinioni binomiali tramite l'equazione (4). Successivamente, tramite l'equazione (9), si possono derivare delle opinioni binomiali a partire dai valori di raccomandazione prodotti dai sistemi di raccomandazione che, nel caso proposto, utilizzano un metodo Collaborative Filtering basato sull'utente. Dopo aver ottenuto le due opinioni binomiali si può utilizzare l'algoritmo di seguito proposto per combinare i risultati. L'algoritmo restituisce solamente il vettore di belief \mathbf{b} del risultato combinato. Infatti è il vettore di belief il dato interessante che determina la qualità della raccomandazione finale. Un vettore di belief alto nel risultato del CasMin indica una buona raccomandazione (punteggi di raccomandazione e reputazione entrambi alti), mentre un valore basso indica una cattiva raccomandazione, ossia l'oggetto non verrà raccomandato all'utente.

2.4.3.1 Pseudocodice

```

1: if  $b^A_X(x) \leq b^B_X(x)$  then
2:    $Swap(\omega^A_X, \omega^B_X)$ 
3: end if
4: if  $u^B_X > (b^A_X(x) - b^B_X(x))$  then
5:    $u^B_X = u^B_X - (b^A_X(x) - b^B_X(x))$ 
6:    $b^B_X(x) = b^A_X(x)$ 
7: else
8:    $b^B_X(x) = b^B_X(x) + u^B_X$ 
9:    $u^B_X = 0$ 
10:   $b^A_X(x) = b^B_X(x)$ 
11: end if
12:  $CasMin = \omega^A_X$ 

```

Algorithm 2: CasMin con opinioni binomiali

2.4.3.2 Esempio

Hotel	Binomial Rep. Score	Binomial Rec. Value	CasMin Advice
Hotel 1	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.1$ $d = 0.16$ $u = 0.2$	$b = 0.3$
Hotel 2	$b = 0.16$ $d = 0.81$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.19$
Hotel 3	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.81$

Tabella 2: Esempio 2 - Risultato CasMin con opinioni binomiali

Applicando l'algoritmo sopra riportato alle due opinioni binomiali riportate in tabella si ottengono i valori di belief in ultima colonna. Si può notare che l'hotel che verrà consigliato dal sistema è il terzo. Infatti, il valore ottenuto dal CasMin è 0.81 che significa che entrambi i sistemi, di reputazione e raccomandazione, hanno restituito buoni punteggi per l'hotel 3. Nel caso dell'hotel 1 si può vedere come, nonostante ci sia un buon punteggio di reputazione (0.81), l'operatore CasMin restituisca un risultato basso (0.3). Questo perché il valore restituito dal sistema di raccomandazione per l'hotel 1 è basso (0.1). Per concludere, nel caso dell'hotel 2 si può vedere come, nonostante ci sia un buon valore di raccomandazione (0.7), l'operatore CasMin restituisca un risultato basso (0.19). Questo perché il punteggio restituito dal sistema di reputazione per l'hotel 2 è basso (0.16).

2.5 Conclusioni

Da quando i sistemi di raccomandazione e reputazione sono a supporto delle decisioni si è pensato che la combinazione degli stessi potesse produrre risultati migliori rispetto ai sistemi isolati. Però, si è compreso che la combinazione dei due sistemi è complessa. Infatti, si è dovuti ricorrere alla teoria della logica soggettiva per rendere omogenei i due risultati.

Con questo breve articolo, il gruppo ha riproposto l'algoritmo di Josang e ha proposto una nuova versione dell'algoritmo che funziona con le opinioni binomiali. Il gruppo ha dovuto creare un nuovo algoritmo perché è risultato difficile ricavare opinioni multinomiali dai sistemi di raccomandazione; per cui risulta difficile applicare l'algoritmo di Josang nella realtà. Il gruppo ritiene che la soluzione proposta sia migliore per un utilizzo con dati reali, e cercherà di provare questa affermazione implementando l'algoritmo in Python e testandolo con dataset reali trovati nel web.

3 Una nostra alternativa

3.1 Introduzione

Il gruppo ha tuttavia pensato ad un'alternativa alla soluzione proposta da Josang e i suoi colleghi per la combinazione dei due sistemi.

Si propone di utilizzare il formalismo dei vincoli soft (SCSP) per combinare le preferenze espresse dai due tipi di sistemi, quello di raccomandazione e quello di reputazione. Per modellare le preferenze sui vincoli è stata adottata una soluzione proposta in un articolo di ricerca (From soft constraints to bipolar preferences: modelling framework and solving issues, vedi bibliografia).

3.2 Descrizione

I vincoli soft vengono modellati utilizzando il formalismo delle preferenze positive, dove la struttura della preferenza positiva sarà così modellata:

$$(P, +_P, \times_P, \perp_P, \top_P)$$

dove:

- P è il set di dati dove $\perp_P, \top_P \in P$;
- $+_P$ rappresenta l'operatore di confronto;
- \times_P rappresenta l'operatore di combinazione;
- \perp_P rappresenta la minima preferenza esprimibile;
- \top_P rappresenta la massima preferenza esprimibile;

Per l'applicazione al caso analizzato si utilizzerà la seguente struttura:

$$(R^+, max, +, 0, +\infty)$$

Quindi la minima preferenza che può essere assegnata è 0. Le preferenze saranno aggregate attraverso la somma e saranno successivamente comparate attraverso l'operatore di massimo.

I vincoli saranno ottenuti mettendo in relazione le entità facenti parte del sistema e collegandole tra loro utilizzando i risultati restituiti dai due sistemi (reputazione e raccomandazione) nel seguente modo:

1. I **valori di belief ottenuti dal sistema di reputazione** vengono associati ad ogni vincolo.

Potrebbe essere (come nel caso dell'esempio) che un'entità in un sistema non abbia un punteggio di reputazione perché non viene considerata dal sistema (ad esempio nel caso di utenti che votano degli hotel, il sistema potrebbe tracciare solo i voti che gli utenti danno agli hotel e non viceversa) ed in questo caso il valore di reputazione assegnato a questa entità deve essere uguale per tutto il dominio (nel caso analizzato 1). È bene considerare che sarà possibile modificare questo valore in futuro, nel momento in cui si riesca ad ottenere punteggi di reputazione anche per quelle entità che prima non venivano valutate.

2. I **valori di belief ottenuti dal sistema di raccomandazione** vengono utilizzati come valore di preferenza per le due entità.

3.3 Sviluppo

Una volta costruiti i vincoli tra le entità utilizzando i valori di belief dei sistemi di raccomandazione e reputazione sarà possibile capire qual è l'associazione migliore che massimizza la preferenza tra le entità coinvolte.

Per ottenere il risultato su n vincoli, basterà combinare le preferenze tra le n entità attraverso l'operatore somma; la preferenza migliore sarà quella massima.

3.4 Esempio

Si fornisce qui di seguito un esempio che utilizza gli stessi valori contenuti negli esempi precedenti, ma in questo caso si utilizza solo il valore di *belief* ottenuto dai sistemi di reputazione (rep) e raccomandazione (rec). I valori di raccomandazione fanno riferimento all'utente *User 1*, su cui poi si baserà l'esempio per predire quale sarà l'hotel da raccomandare.

Hotel	Binomial Rep. Score	Binomial Rec. Score
Hotel 1	b = 0.81	b = 0.1
Hotel 2	b = 0.16	b = 0.7
Hotel 3	b = 0.81	b = 0.7

Tabella 3: Dati risultati da rep. and rec. system

Che rappresentati graficamente come vincoli diventano:

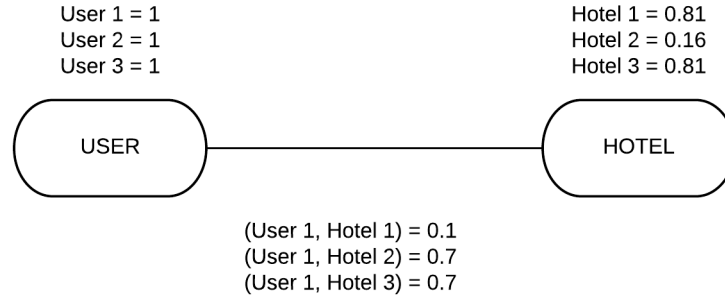


Figura 1: Esempio con vincoli soft

Quindi, effettuando la combinazione dei valori i risultati sono:

$$\begin{aligned}(User1, Hotel1) &= 1 + 0.1 + 0.81 = 1.91 \\(User1, Hotel2) &= 1 + 0.7 + 0.16 = 1.86 \\(User1, Hotel3) &= 1 + 0.7 + 0.81 = 2.51\end{aligned}$$

Giunti a questo punto si utilizza l'operatore di ottimizzazione (*operatore di massimo*) per ottenere il risultato che, in questo caso, è la combinazione *User 1 & Hotel 3*, che di fatto rappresenta la migliore scelta per l'utente 1.

3.5 Conclusioni

A differenza dell'algoritmo individuato da Josang, la soluzione proposta non dà priorità alle preferenze che hanno un voto "medio/alto" per entrambi i sistemi (quello di reputazione e quello di raccomandazione), bensì ottimizza il soddisfacimento in generale. In molti casi la soluzione proposta dallo studio di Josang e la nostra alternativa riportano i medesimi risultati, ma potrebbero esserci casi in cui questo non accade. Sugeriamo di consultare la sezione riguardo alla nostra implementazione per avere ulteriori riscontri.

4 La nostra implementazione

Il gruppo ha implementato degli script in Python per la realizzazione della logica applicativa e un'applicazione web per la visualizzazione dei risultati.

4.1 Script in Python

4.1.1 Reperimento del codice

Il codice sorgente dell'applicazione è reperibile sulla repository GitHub al seguente link: <http://github.com/bmxitalia/ProgettoIA>.

4.1.2 Dataset di input

Il gruppo, dopo un'approfondita ricerca, ha selezionato un dataset di utenti e hotel che ha ritenuto opportuno per il progetto. Tale dataset si trova all'interno della cartella "source". Il dataset contiene recensioni di utenti per degli hotel e i dati utili sono i seguenti:

- **name**: nome dell'hotel;
- **reviews.rating**: punteggio da 1 a 10 attribuito dall'utente all'hotel;
- **reviews.username**: nome utente dell'utente recensore.

4.1.3 Logica applicativa

Il gruppo ha realizzato due script PHP:

1. ***tableConstruct.py***: tale script si occupa di:
 - prelevare i dati descritti nella sezione precedente dal dataset;
 - inserire tali dati in dizionari opportunamente costruiti;
 - manipolare i dati per ricavare i punteggi di reputazione di ogni hotel;
 - manipolare i dati per ricavare i valori di raccomandazione per ogni coppia utente/hotel;
 - convertire i punteggi di reputazione in opinioni soggettive binomiali;
 - convertire i valori di raccomandazione in opinioni soggettive binomiali;
 - memorizzare le opinioni soggettive ottenute in dei dataset (*/output/rep_bin.csv* e */output/rec_bin.csv*).
2. ***computeOutput.py***: tale script si occupa di:
 - prelevare i dati dai dataset contenenti le opinioni soggettive;
 - inserire tali dati in dizionari opportunamente costruiti;

- combinare le opinioni soggettive dei punteggi di reputazione con le opinioni soggettive dei valori di raccomandazione con applicazione di CasMin e vincoli soft;
- memorizzare i risultati ottenuti in un dataset finale (*/output/finale.csv*).

4.1.4 Dataset di output

I dataset di output sono contenuti nella cartella “output” e sono i seguenti:

- ***rec_bin.csv***: contiene i valori di raccomandazione, per ogni coppia utente/hotel, espressi come opinioni soggettive binomiali;
- ***rep_bin.csv***: contiene i punteggi di reputazione, per ogni hotel, espressi come opinioni soggettive binomiali;
- ***casmin.csv***: contiene i consigli CasMin dati dalla combinazione dei valori di raccomandazione e dei punteggi di reputazione, per ogni coppia utente/hotel, espressi come opinioni soggettive binomiali;
- ***vincoliSoft.csv***: contiene le somme risultanti dalla combinazione dei valori di raccomandazione e dei punteggi di reputazione, utilizzando i vincoli soft, per ogni coppia utente/hotel;
- ***finale.csv***: contiene, per ogni utente, l’hotel consigliato da CasMin e quello raccomandato dai vincoli soft.

4.2 Applicazione web

4.2.1 Reperimento dell’applicazione web

L’applicazione web è disponibile al seguente link: <http://progettoiaunipd.altervista.org>.

Inoltre il sorgente dell’applicativo web è reperibile anch’esso nella repository GitHub messa a disposizione: <https://github.com/bmxitalia/ProgettoIA>.

4.2.2 Tabelle sorgenti

<http://progettoiaunipd.altervista.org/sorgenti.php>

In questa pagina è possibile visualizzare a video le tabelle sorgenti relative agli hotel e agli utenti. Sarà dunque possibile visualizzare tutti gli hotel e gli utenti presenti nel dataset utilizzato.

4.2.3 Risultati

In questa serie di pagine è possibile visualizzare tutti i risultati ottenuti attraverso l’algoritmo di CasMin e i vincoli soft.

4.2.3.1 Combinazione finale

<http://progettoiaunipd.altervista.org/final.php>

In questa pagina è possibile avere un riepilogo generale di confronto tra i due metodi analizzati, ossia quello di CasMin e quello dei vincoli soft. La tabella presentata mostrerà per ogni utente quale sarebbe l'hotel "raccomandato" con CasMin e quale sarebbe quello "raccomandato" utilizzando i vincoli soft.

4.2.3.2 Algoritmo di CasMin

<http://progettoiaunipd.altervista.org/casmin.php>

In questa pagina è possibile visualizzare i risultati ottenuti dagli script in python applicando l'algoritmo di CasMin. La tabella presentata mostrerà per ogni utente qual è l'hotel che sarebbe "raccomandato" e con quale valore binomiale.

4.2.3.3 Vincoli soft

<http://progettoiaunipd.altervista.org/scsp.php>

In questa pagina è possibile visualizzare i risultati ottenuti dagli script in python applicando i vincoli soft. La tabella presentata mostrerà per ogni utente qual è l'hotel che sarebbe "raccomandato" e con quale risultato di combinazione.

4.2.4 Ricerca dettagliata

<http://progettoiaunipd.altervista.org/research.php>

In questa pagina è possibile fare una ricerca mirata sui risultati ottenuti. Sarà possibile scegliere un utente e visualizzare in dettaglio quale sarebbe l'hotel consigliato utilizzando CasMin e quale sarebbe quello consigliato utilizzando i vincoli soft.

5 Bibliografia

1. *Combining Recommender and Reputation Systems to Produce Better On-line Advice*, Audun Jøsang, Guibing Guo, Maria Silvia Pini, Francesco Santini, Yue Xu
2. *Recommendation systems: Principles, methods and evaluation*, F.O.Isinkaye Y.O.Folajimi B.A.Ojokoh
3. *Interactive Reputation Systems*, Johannes Sanger Gunther Pernul
4. *From soft constraints to bipolar preferences: modelling framework and solving issues*, Stefano Bistarelli, Maria Silvia Pini, Francesca Rossi and K. Brent Venable
5. *Slide corso 'Intelligenza artificiale a.a. 2017/2018', Universit degli Studi di Padova - Informatica*, Maria Silvia Pini