UNIVERSITÀ DEGLI STUDI DI PADOVA
FACOLTÀ DI INGEGNERIA
Corso di Laurea in Ingegneria dell'Informazione

SISTEMI DI REPUTAZIONE E RACCOMANDAZIONE INTELLIGENTI

Laureando: Paolo Bonaventura

Relatore: Dott.ssa Maria Silvia Pini

Anno accademico 2012-2013

Indice

Sommario **1**

1 Introduzione **3**

2 Sistemi di Reputazione **7**
2.1 Sistemi di Reputazione Centralizzati 9
2.2 Sistemi di Reputazione Distribuiti 11
2.3 Motori di Calcolo della Reputazione 12
2.3.1 Somma Semplice dei Voti 13
2.3.2 Media dei Voti 14
2.3.3 Sistemi Bayesiani 15
2.3.4 Modelli Discreti 17
2.3.5 Modelli di Belief 17
2.3.6 Modelli di Flusso 19
2.4 Sistemi di Reputazione nella Computer Security 20

3 Sistemi di Raccomandazione **23**
3.1 Metodi Basati sul Contenuto 24
3.2 Metodi Collaborativi 27
3.2.1 Algoritmi Basati sulla Memoria 27
3.2.2 Algoritmi Basati sui Modelli 29
3.3 Problematiche dei Sistemi Content-based e Collaborativi 30
3.4 Metodi Ibridi 31
3.4.1 Estendere i Sistemi di Raccomandazione 33
3.5 Efficienza 38

4 Integrare Sistemi di Reputazione e Raccomandazione **39**

5 Conclusioni **43**

6 Sviluppi Futuri **45**

Sommario

Il lavoro svolto in questa tesi si concentra sull'analisi dei sistemi di reputazione e quelli di raccomandazione, ossia due tipi di sistemi di supporto alle decisioni che sono in grado di fornire agli utenti delle informazioni utili nei processi decisionali. Vengono studiati i tipi di sistemi più diffusi, descritti nella letteratura, facendo attenzione a quali sono i loro pro e contro in modo da poter delineare come questi potrebbero essere migliorati. A questo scopo viene anche descritto un metodo utilizzato per integrare i sistemi di reputazione e raccomandazione con il quale si possono ottenere dei risultati molto più accurati. L'obiettivo risulta quindi essere quello di presentare una panoramica dell'attuale stato dell'arte di questo settore, fornendo degli spunti per migliorare i sistemi dell'odierna generazione.

Capitolo 1

Introduzione

I sistemi di reputazione, così come quelli di raccomandazione, hanno avuto un forte sviluppo in quest'ultimo decennio, in seguito alla riuscita integrazione di tali sistemi in applicazioni commerciali, soprattutto nel mondo dell'E-commerce. Il motivo del loro successo risiede nel fatto che sono in grado di aiutare un utente a decidere con chi effettuare una transazione e cosa acquistare.

In questa tesi vengono analizzati tali sistemi ponendo particolare attenzione ai loro vantaggi e alle loro limitazioni, con lo scopo di determinare in che modo questi potrebbero essere migliorati.

Un sistema di reputazione [15] ha la potenzialità di offrire le basi per prendere una decisione in modo più consapevole, fornendo ai membri del sistema delle informazioni riguardanti l'affidabilità degli altri utenti. L'idea di fondo di questo tipo di sistemi consiste nel permettere agli utenti coinvolti in una transazione di valutare la propria controparte, fornendo un giudizio che riflette in che misura quell'utente è affidabile. In questo modo si possono utilizzare tutti i voti assegnati dai membri del sistema per calcolare il punteggio di reputazione di un dato utente, reso disponibile a tutta la comunità o che si può consultare su richiesta. Così facendo gli utenti possono sfruttare questo punteggio per decidere se effettuare la transazione che avevano previsto, oppure per scegliere con chi effettuare una futura transazione. Quindi queste applicazioni hanno un effetto positivo nel contesto in cui vengono inserite anche perché inducono gli utenti a comportarsi nel miglior modo possibile, per non veder intaccata la loro reputazione.

I sistemi di reputazione possono essere classificati in *sistemi centralizzati* e *sistemi distribuiti* in base alla loro architettura e si differenziano per il modo in cui i voti e i punteggi di reputazione vengono raccolti e distribuiti agli utenti. Nei sistemi centralizzati, in seguito ad ogni

transazione, gli utenti forniscono un voto per dare un'informazione sull'affidabilità dell'altro utente. Questi voti vengono raccolti da un *centro di reputazione* e vengono usati per calcolare dei *punteggi di reputazione* da attribuire ai vari utenti. In un sistema distribuito invece i voti vengono memorizzati dai singoli utenti, che possono essere visti come dei nodi all'interno di una rete. Quando un utente vuole effettuare una transazione con qualcun altro deve prima cercare i nodi che contengono le informazioni relative all'utente desiderato e poi richiedere il suo punteggio di reputazione, calcolato dai singoli nodi. Per calcolare i punteggi di reputazione esistono varie metodologie, la più semplice delle quali è la somma dei voti (adottata nel feedback forum di eBay) [20]: tenendo traccia del numero di voti positivi p e il numero di voti negativi n assegnati all'utente u, il punteggio di reputazione per u viene calcolato come differenza tra p e n, cioè $p - n$. Le altre tecniche più usate sono: la media dei voti, sistemi Bayesiani [14], modelli discreti [1] (dove la reputazione di un utente può assumere solo dei valori prestabiliti, ad esempio "molto affidabile", "affidabile", "inaffidabile", o "molto inaffidabile" attribuiti all'utente utilizzando delle tabelle di ricerca), modelli di belief [11, 12, 25] (che utilizzano la teoria del belief per determinare in che misura si crede che un utente sia affidabile) e modelli di flusso (che calcolano iterativamente la reputazione, come nel caso di Google PageRank) [17].

Per quanto riguarda i sistemi di raccomandazione [3] invece, di cui l'esempio probabilmente più famoso è Amazon, sono in grado di suggerire agli utenti gli oggetti che presumibilmente saranno di loro interesse. In pratica utilizzano le informazioni ricevute direttamente dall'utente, riguardo ai prodotti ai quali è interessato, per capire quali sono i suoi gusti e raccomandare l'oggetto che più si adatta alle sue necessità. Altrimenti il sistema costruisce un database di tutti i clienti e le loro preferenze, per poi determinare quali sono le persone più simili (neighbors) all'utente al quale si vuole fornire la raccomandazione, cioè quelle che hanno dei gusti simili. Quindi vengono raccomandati gli oggetti che piacciono ai *neighbors*, poiché molto probabilmente questi piaceranno anche all'utente. I sistemi di raccomandazione sono usati non solo in siti di E-commerce, ma anche in altri tipi di applicazioni, come ad esempio sistemi che raccomandano notizie nei newsgroups, oppure sistemi che raccomandano libri o film.

Per fornire una raccomandazione il sistema deve stimare il voto che presumibilmente l'utente darebbe ad un particolare oggetto e poi determinare l'oggetto che ha ottenuto il punteggio stimato più alto. In base all'approccio usato nel derivare la stima dei voti i sistemi di raccoman-

dazione si possono classificare in: sistemi basati sul contenuto (content-based), sistemi collaborativi e sistemi ibridi. Nei sistemi del primo tipo i voti stimati per l'utente u vengono calcolati sulla base dei voti forniti dallo stesso utente u per oggetti simili a quello che si vuole raccomandare. Nei sistemi collaborativi, detti anche di *filtraggio collaborativo*, la stima avviene sulla base dei voti che altri utenti della comunità hanno fornito per lo stesso oggetto che si vuole raccomandare. I sistemi ibridi invece combinano i sistemi sopra citati allo scopo di evitare alcune problematiche che affliggono i primi due sistemi. Tra queste si può citare il problema dei nuovi utenti: poiché il sistema formula le raccomandazioni sulla base delle preferenze assegnate dall'utente, non è in grado di fornire raccomandazioni accurate ad un utente che è appena entrato a far parte del sistema e quindi non ha ancora espresso le sue preferenze.

Nonostante il loro forte sviluppo i sistemi di reputazione e quelli di raccomandazione necessitano ancora di miglioramenti. Si stanno infatti cercando dei metodi per rendere i sistemi di raccomandazione più accurati e meno invasivi, per applicarli anche a sistemi più complessi (ad esempio un sistema di raccomandazione per viaggi, nel quale oltre alle preferenze dell'utente si possono considerare altre informazioni variabili come la situazione metereologica o anche la situazione economica e politica della località che si vuole raccomandare). In quest'ottica si inserisce un metodo di integrazione dei sistemi di reputazione e raccomandazione [13] con il quale si possono produrre delle raccomandazioni molto più accurate. Con questo metodo vengono raccomandati gli oggetti ai quali gli utenti sono potenzialmente interessati (utilizzando il sistema di raccomandazione) privilegiando però gli oggetti che hanno una alta reputazione, ottenendo così delle raccomandazioni più affidabili.

Inoltre si sta cercando di rendere più robusti i sistemi di reputazione, in modo da mitigare i possibili attacchi da entità esterne per modificare a proprio vantaggio i punteggi di reputazione.

In questa tesi vengono descritti i metodi che possono essere utilizzati per calcolare i punteggi di reputazione e vengono anche analizzate alcune delle applicazioni più comuni che fanno uso di questi sistemi (come eBay, Amazon, Google PageRank). Inoltre viene fornita una panoramica sullo stato dell'arte dei sistemi di raccomandazione, facendo attenzione alle loro limitazioni e cercando di fornire alcune soluzioni per superarle. Viene anche illustrato un metodo recentemente studiato per integrare i sistemi di reputazione e raccomandazione in modo da poter ottenere delle raccomandazioni molto più accurate e affidabili.

La tesi è strutturata nel seguente modo: nel Capitolo 2 vengono descritti i sistemi di reputazione e i metodi comunemente utilizzati per definire i punteggi di reputazione per gli utenti del sistema; nel Capitolo 3 vengono presentati i sistemi di raccomandazione, classificandoli in base all'approccio usato per formulare la raccomandazione in tre classi diverse: sistemi basati sul contenuto, sistemi collaborativi e sistemi ibridi. Vengono inoltre descritte le problematiche che questi sistemi devono affrontare, cercando di presentare alcune soluzioni per superarle. Nel Capitolo 4 viene introdotto un metodo da poco sviluppato che permette di integrare i sistemi di reputazione e quelli di raccomandazione allo scopo di formulare delle migliori raccomandazioni nei servizi online. Infine i Capitoli 5 e 6 contengono rispettivamente le conclusioni e alcuni potenziali sviluppi futuri di questo tipo di sistemi.

Capitolo 2

Sistemi di Reputazione

Un sistema di reputazione è una struttura in grado di raccogliere informazioni che gli utenti forniscono riguardo alle interazioni avvenute con altri utenti, e di utilizzarle per calcolare un punteggio che determina la reputazione di questi utenti. Tali punteggi possono quindi essere consultati da tutti per decidere se fidarsi di un determinato utente ed intraprendere una transazione con lui, oppure per determinare chi è l'utente più attendibile per effettuare la transazione desiderata.

Per descrivere un sistema di reputazione è necessario prendere in considerazione le nozioni di fiducia e reputazione, risulta perciò conveniente dare una definizione per questi concetti che verranno poi utilizzati. È possibile definire il concetto fiducia in due modi differenti [15].

1. *Considerati due individui A e B, la fiducia è la probabilità con cui A si aspetta che B compia una determinata azione, dalla quale A può beneficiarne.*

Questa definizione venne fornita da Gambetta [9] e può essere interpretata come affidabilità, riferita ad una persona o una cosa.

2. *La fiducia è la misura in cui una persona è disposta a dipendere da qualcosa o qualcuno in una data situazione, con una sensazione di relativa sicurezza, nonostante possano esserci delle conseguenze negative.*

Questa è la definizione che verrà adottata nel resto della tesi e risulta essere una nozione più generale che introduce dei nuovi concetti, tra i quali: *dipendenza* dall'entità in cui si ripone la propria fiducia, *utilità* (nel senso che una relazione tra due utenti può avere esiti positivi o anche negativi), e *attitudine al rischio* ossia il fatto che un utente è disposto ad accettare una situazione che per lui può essere svantaggiosa.

Un altro concetto legato alla nozione di fiducia è quello di *reputazione*: la reputazione di un soggetto (persona, istituzione, azienda etc) è la

considerazione o la stima di cui questo soggetto gode nella società.

Quindi i due concetti di fiducia e reputazione sono tra loro differenti: il primo infatti è un concetto più soggettivo, basato principalmente sull'esperienza personale, mentre il secondo viene derivato dalla comunità appartenente ad una rete sociale, e può essere considerato come una misura collettiva della fiducia. La differenza tra i due termini si può comprendere nelle seguenti frasi:

(1) "Mi fido di te per la tua buona reputazione".

(2) "Mi fido di te nonostante la tua cattiva reputazione".

Nella prima frase l'utente basa la sua fiducia sulla reputazione che l'altro utente ha, mentre nella seconda l'utente dimostra di avere qualche conoscenza personale riguardo l'altro utente, ad esempio ottentuta per esperienza diretta, e questa viene ritenuta dominante rispetto alla reputazione che una persona può avere. Questo riflette il fatto che la fiducia fondamentalmente è incentrata su un giudizio personale e soggettivo, basato su molti fattori, alcuni dei quali hanno una maggiore importanza. Ad esempio, solitamente l'esprienza personale è piu importante rispetto alle raccomandazioni fatte da altri utenti, ma nel caso in cui l'utente non abbia mai avuto esperienze dirette la fiducia può basarsi esclusivamente su queste raccomandazioni.

Gli obiettivi che la ricerca nei sistemi di reputazione cerca di soddisfare sono principalmente i seguenti:

- trovare nuove informazioni ed elementi che possono essere utilizzati nelle applicazioni online per calcolare una misura di fiducia e reputazione;
- creare dei sistemi efficienti per raccogliere queste informazioni e aiutare gli utenti a prendere decisioni sulle future transazioni, migliorando così la qualità delle applicazioni commerciali online.

Secondo ciò che afferma Resnick *et al.* [21] un sistema di reputazione deve avere le seguenti proprietà:

- gli elementi di tale sistema (gli utenti che ne fanno parte) devono essere longevi, nel senso che devono essere presenti nel sistema da molto tempo, in modo da poter sempre avere delle future interazioni e conoscere l'esito di quelle precedenti,

- i voti riguardanti le interazioni correnti devono essere raccolti e poi resi pubblici ai membri del sistema,
- i voti riguardanti le interazioni precedenti devono influenzare e guidare le interazioni correnti.

La prima proprietà serve per evitare i cambiamenti di identità, con i quali un utente può eliminare tutte le connessioni con le sue passate transazioni, allo scopo di cancellare la propria reputazione. La seconda proprietà principalmente riguarda il modo in cui il sistema acquisisce i voti assegnati dagli utenti, e dipende anche dalla volontà degli stessi utenti di fornire tali voti, in quanto un utente potrebbe anche non volere esprimere la propria opinione (per questo motivo è utile introdurre una qualche forma di incentivo). Invece la terza proprietà riguarda fondamentalmente l'efficienza del sistema e la sua utilizzabilità da parte degli utenti.

Principalmente i sistemi di reputazione possono essere classificati in due categorie, in base alla loro architettura, e differiscono per il modo in cui i voti e i punteggi di reputazione vengono raccolti e distribuiti tra i vari utenti. Queste due categorie sono: *architetture centralizzate* e *architetture distribuite*.

2.1 Sistemi di Reputazione Centralizzati

In un sistema di reputazione centralizzato, le informazioni riguardanti le transazioni effettuate con un certo partecipante vengono raccolte sotto forma di voto (un punteggio) fornito dagli utenti della comunità che hanno avuto un'esperienza diretta con questo partecipante. Questi voti vengono raccolti da un'autorità centrale (il *centro di reputazione*) che poi genera un punteggio di reputazione da attribuire ad ogni utente, rendendolo pubblicamente visibile ai membri del sistema. A questo punto gli utenti possono utilizzare tali voti per decidere se e con chi effettuare una transazione. L'idea alla base di ciò è che una transazione con un partecipante con una buona reputazione tenderà ad essere favorevole per entrambe le due parti, ossia acquirente e venditore, mentre una transazione con un utente caratterizzato da una bassa reputazione potrebbe avere delle conseguenze negative. La Figura 2.1 illustra schematicamente la struttura di un possibile sistema di reputazione centralizzato, in cui gli utenti A e B sono intenzionati ad effettuare una transazione tra loro.

Il centro di reputazione raccoglie i voti degli utenti A, B, C, D, E, F, G in seguito ad ogni transazione, ad esempio quella tra A e C oppure D

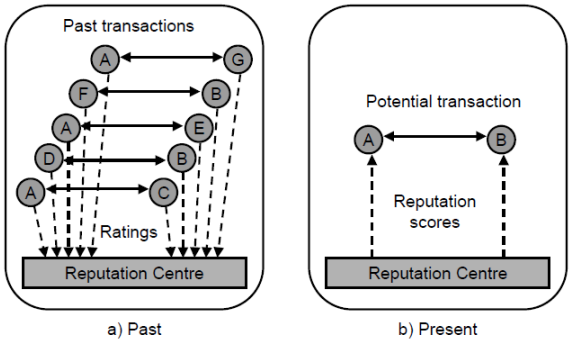


Figura 2.1: Struttura di un sistema di reputazione centralizzato.

e B. Questi voti vengono poi utilizzati per calcolare i punteggi di reputazione da attribuire ai vari utenti, aggiornandoli dopo ogni transazione in funzione dei voti ricevuti. A questo punto, supponendo che gli utenti A e B vogliano effettuare una transazione tra loro, questi possono utilizzare il centro di reputazione per ottenere il punteggio di reputazione associato all'altro utente, derivando così un'informazione relativa alla sua affidabilità.

In un sistema di reputazione centralizzato si possono distinguire principalmente due aspetti fondamentali [15]:

- Un *protocollo di comunicazione centralizzato* con il quale l'autorità centrale è in grado di raccogliere i voti forniti dagli utenti a seguito di una transazione, oltre a permettere ai membri di ottenere i punteggi di reputazione dell'utente desiderato,

- Un *motore di calcolo della reputazione*, cioè un metodo utilizzato dall'autorità centrale per calcolare i punteggi di reputazione, basandosi sui voti che gli utenti hanno fornito, e in alcuni casi anche su altre informazioni, come ad esempio la reputazione degli altri utenti.

Ad esempio, supponendo di avere tre utenti A, B e C e di voler calcolare la reputazione di A, i voti forniti dagli altri membri possono essere pesati tenendo conto della reputazione di questi ultimi. In questo modo se l'utente B, con una bassa reputazione, fornisce un voto molto basso per A (che quindi verosimilmente non rispecchierà la vera affidabilità di A) mentre C, con alta reputazione, fornisce un voto alto, il voto di B verrà considerato con un peso minore a quello di C, perciò la reputazione di A risulta essere comunque alta, nonostante il voto molto basso fornito da B.

2.2 Sistemi di Reputazione Distribuiti

In alcuni casi può essere conveniente avere un'architettura di tipo distribuito, cioè una struttura in cui non c'è un'autorità centrale che raccoglie voti e rende disponibili i punteggi. Questa struttura è come una rete in cui ci sono vari nodi che fungono da centro di raccolta, ai quali si possono fornire i voti riguardanti le transazioni effettuate; oppure ogni membro costituisce un nodo all'interno della rete, e raccoglie e registra le proprie opinioni sulle interazioni avvenute con gli altri utenti, per poi fornire queste informazioni su richiesta a coloro che desiderano conoscere la reputazione di un particolare utente. Quindi l'utente che vuole effettuare una transazione con un determinato membro, deve in primo luogo trovare l'utente (nodo della rete) o il maggior numero possibile di utenti che hanno avuto un'esperienza diretta con quel membro, che quindi hanno raccolto delle informazioni relative ad esso, e poi richiedere queste informazioni sotto forma di punteggio di reputazione. Questo tipo di struttura può essere rappresentato come nella Figura 2.2: a seguito delle transazioni avvenute tra gli utenti A e C, D e B, A e E, F e B, A e G, ogni nodo (utente) memorizza la propria opinione (voto) relativa all'altro utente coinvolto nella transazione (ad esempio A memorizza i voti che lui ha attribuito a C, E e G). Se a questo punto A è interessato ad effetuare una transazione con B, A dovrà richiedere agli utenti che hanno avuto un'esperienza diretta con B (cioè D e F) le loro opinioni (punteggi di reputazione) e allo stesso modo B richiederà a C, E e G i punteggi di reputazione che loro hanno assegnato ad A.

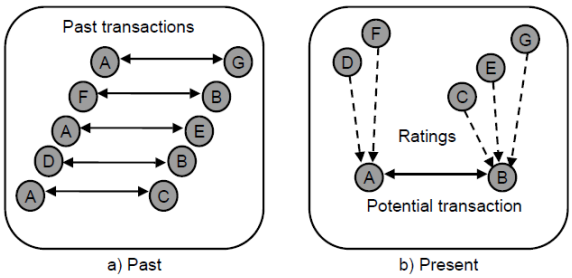


Figura 2.2: Struttura di un sistema di reputazione distribuito.

Come per un sistema di reputazione centralizzato, si possono definire due aspetti fondamentali di un sistema di reputazione distribuito:

- Un *protocollo di comunicazione distribuito* necessario per ottenere i voti dagli altri utenti del sistema,

- Un *metodo di calcolo della reputazione* con il quale ogni singolo utente del sistema può calcolare i punteggi di reputazione degli altri membri, basandosi sui voti che sono stati ricevuti.

In un sistema distribuito ogni membro deve sia raccogliere i voti, sia determinare i punteggi di reputazione per gli altri membri. Poiché può risultare molto dispendioso, o addirittura impossibile, considerare i voti derivati da *tutte* le interazioni con un particolare utente, spesso il punteggio di reputazione viene calcolato basandosi solamente su un sottoinsieme di voti.

Un sistema che rappresenta bene questa struttura è una rete Peer-to-Peer (P2P) [15]. In una rete di questo tipo ogni nodo (utente) svolge sia il compito di client sia quello di server, per questo viene anche chiamato *servent*. Nell'utilizzo di una rete P2P si possono distinguere due fasi: la fase di ricerca e la fase di download. La prima consiste nella ricerca del servent nel quale risiede la risorsa voluta. Questa fase può essere implementata con funzioni distribuite o anche centralizzate, ad esempio utilizzando un server di ricerca che tiene traccia di tutti i nodi della rete e delle informazioni che questi hanno raccolto. Dopo aver trovato il servent che contiene le informazioni volute si passa alla fase di download, con la quale queste informazioni vengono trasferite da un nodo all'altro.

Con questo tipo di reti però vengono introdotte delle nuove problematiche relative alla sicurezza degli utenti. Infatti una rete P2P può essere utilizzata per mettere in circolazione dei software dannosi (ad esempio un virus) tra i vari membri della rete.

Introducendo un sistema di reputazione all'interno di una rete P2P si può mitigare questo problema. Infatti condividendo informazioni riguardo agli altri utenti della rete si può determinare quali sono i servents più affidabili e che offrono le risorse di miglior qualità e allo stesso modo rintracciare anche gli utenti inaffidabili e quelli sospetti.

2.3 Motori di Calcolo della Reputazione

I punteggi di reputazione possono essere calcolati in diversi modi, e basandosi su molteplici fattori, ad esempio sull'esperienza personale o su raccomandazioni di altri utenti. Adottando le definizioni utilizzate nelle teorie economiche, le informazioni derivate dall' esperienza personale vengono indicate con il termine *informazioni private* (private informations), mentre per quelle pubblicamente disponibili, ottenute da altri utenti, si utilizza il termine *informazioni pubbliche* (public informations). Tipica-

mente un sistema di reputazione è basato sulle informazioni pubbliche, dato che lo scopo è quello di riflettere le opinioni dei vari utenti di una comunità. I metodi per ottenere un puteggio di reputazione sono numerosi; di seguito verranno elencati quelli piu usati nel settore delle applicazioni commerciali e di maggiore importanza.

2.3.1 Somma Semplice dei Voti

Il metodo più semplice usato per calcolare i punteggi di reputazione è utilizzare la somma semplice dei voti: considerato un utente che ha ricevuto "p" voti positivi e "n" voti negativi, il punteggio di reputazione attribuito a questo utente si ottiene calcolando la differenza tra i voti positivi e quelli negativi, cioè " p - n " .

Questo metodo di calcolo è stato adottato dal forum di reputazione di eBay (Feedback Forum) [20], con il quale venditori e acquirenti possono esprimere un voto positivo, neutro, o negativo (+1, 0, -1) dopo aver terminato una transazione, oltra ad avere la possibilità di lasciare un commento riguardante la qualità dell'operazione. Questo forum è un sistema di reputazione centralizzato con il quale vengono raccolti tutti i voti assegnati dai vari utenti e viene reso disponibile il punteggio di reputazione (Feedback) di ogni utente sotto forma di somma totale dei voti ricevuti (numero di voti positivi meno numero di voti negativi). È inoltre possibile vedere i voti positivi e negativi separatamente relativi a tre diverse finestre temporali: 1) ultimi sei mesi, 2) ultimo mese, 3) ultima settimana, allo scopo di fornire agli utenti delle informazioni relative al comportamento più recente dell'utente in questione.

Il vantaggio di questo metodo di calcolo consiste nella sua semplicità: oltre ad essere semplice da implementare, ogni utente è in grado di comprendere il principio alla base del calcolo del punteggio di reputazione. Ci sono però anche alcuni svantaggi: in primo luogo fornisce una scarsa indicazione sulla reputazione dei partecipanti, ad esempio un utente con 100 voti positivi e 10 negativi viene reputato allo stesso modo di un'altro utente con 90 voti positivi e 0 negativi, mentre intuitivamente dovrebbe essere considerato meno affidabile. Un altro problema che sistemi di questo tipo devono affrontare è quello dei voti fraudolenti, ad esempio quando qualcuno vota ripetutamente in modo positivo un utente allo scopo di aumentare il suo punteggio di reputazione. Infatti si possono creare delle finte transazioni con il solo intento di dare un voto positivo alla controparte, problema che è stato risolto da eBay introducendo una tassa da pagare per ogni inserzione effettuata nel sito. Tuttavia è possibile che in seguito ad una transazione autentica un utente fornisca

un voto ingiusto (ad esempio un voto negativo anche se la transazione è stata di ottima qualità), e questo è un problema che non può essere evitato.

2.3.2 Media dei Voti

Un metodo di calcolo della reputazione leggermente più avanzato consiste nella media dei voti: il punteggio di reputazione di un particolare utente viene calcolato come voto medio di tutti i voti che quell'utente ha ricevuto.

Anche questo è un metodo utilizzato in molti siti web commerciali, tra i quali si può citare Amazon. Inizialmente Amazon era una libreria online, ora diventata una compagnia di commercio elettronico nella quale i membri possono recensire i prodotti presenti, ad esempio libri, e fornire un voto, compreso nel range da 1 a 5. Il voto complessivo attribuito a quel prodotto è dato dalla media di tutti i voti che sono stati forniti. Si possono dare delle informazioni anche sulle recensioni lasciate dagli altri utenti, votandole come *utili* (voto positivo) o *non utili* (voto negativo). Queste informazioni contribuiranno poi a determinare il rank del recensore, che è influenzato anche da altri fattori, come il numero di altri utenti che hanno votato il recensore come favorito. Questo sistema di recensioni però risulta essere poco sicuro, infatti è facilmente soggetto ad attacchi esterni: ad esempio un editore potrebbe essere interessato a fornire degli incentivi (anche economici) ad alcuni utenti del sistema, per pubblicare delle recensioni favorevoli per i suoi libri. Inoltre anche questo sistema è afflitto dal problema dei voti fraudolenti, in quanto qualsiasi utente può votare senza diventare membro (essere registrato nel sito), per cui si può artificialmente modificare la reputazione di un altro utente fornendo dei voti favorevoli o sfavorevoli. Ad esempio votando in modo negativo le recensioni effettuate da un utente, la reputazione di questo viene abbassata. Contrariamente se si votano positivamente le sue recensioni la reputazione aumenta. È perciò evidente che la reputazione di un utente può essere modificata a proprio piacimento, votando ripetutamente le sue recensioni in modo positivo o negativo. Per cercare di risolvere questo problema, Amazon tiene traccia dei cookies dei vari recensori e permette a questi ultimi di effettuare un solo voto per cookie registrato. Tuttavia cancellando i cookies o utilizzando un'altro computer, lo stesso utente può votare nuovamente lo stesso prodotto.

Ci sono anche modelli di calcolo di questo tipo più avanzati, come ad esempio una media pesata dei voti, dove i singoli voti dati dagli utenti vengono pesati in base alla reputazione dell'utente che vota, al periodo in

cui il voto è stato dato, oppure alla differenza tra il voto fornito e il punteggio corrente. Ad esempio se l'utente B fornisce un voto per l'utente A che è molto minore all'attuale reputazione di A, questo voto verrà considerato con un peso minore rispetto agli altri voti, poichè verosimilmente quel voto non riflette la reale affidabilità di A.

2.3.3 Sistemi Bayesiani

Un altro metodo che permette di calcolare i punteggi di reputazione consiste nell'utilizzo di un sistema Bayesiano. Prendendo in ingresso dei voti di tipo binario, ad esempio positivi o negativi, i punteggi si possono calcolare basandosi sulla funzione di densità di probabilità (PDF) di una distribuzione beta B. Si tratta di una distribuzione di probabilità continua definita su di un intervallo unitario [0, 1] da due parametri α e β. Questa distribuzione è in grado di governare la probabilità p di un processo Bernoulli a posteriori dell'osservazione di $\alpha - 1$ *successi* e $\beta - 1$ *fallimenti*, sapendo che p è a priori uniformemente distribuito in [0, 1]. Il punteggio di reputazione a *posteriori* può essere calcolato combinando il punteggio a *priori* con i nuovi voti assegnati.

La funzione di densità di probabilità di una tale distribuzione è definita come segue:

$$Beta(p|\alpha, \beta) = \frac{p^{\alpha-1}(1-p)^{\beta-1}}{B(\alpha, \beta)} \tag{2.1}$$

dove B(α,β) è la *funzione beta* così definita:

$$B(\alpha, \beta) = \int_0^1 p^{\alpha-1}(1-p)^{\beta-1}\mathrm{d}p \tag{2.2}$$

la quale può anche essere espressa in termini della funzione gamma (Γ) come segue:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \tag{2.3}$$

con $0 \leq \mathrm{p} \leq 1$, α, $\beta > 0$, e aggiungendo le seguenti condizioni:

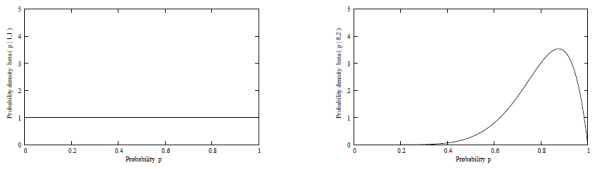
- $\mathrm{p} \neq 0$ se $\alpha < 1$,
- $\mathrm{p} \neq 1$ se $\beta < 1$.

Quindi, definita una PDF beta con parametri α e β, che rappresentano rispettivamente il numero di voti positivi e negativi, il punteggio di reputazione può essere rappresentato sotto forma di valore atteso di

questa PDF, eventualmente accompagnato anche dalla sua varianza. Il valore atteso di una beta PDF è dato da:

$$E[p] = \int_{-\infty}^{\infty} p \cdot Beta(p|\alpha, \beta)\mathrm{d}p = \frac{\alpha}{\alpha + \beta} \tag{2.4}$$

Per convezione, quando non si hanno informazioni riguardo al punteggio di reputazione di un particolare utente, si assume che la distribuzione a *priori* è rappresentata da una beta PDF unforme con parametri $\alpha = 1$ e $\beta = 1$, mentre, dopo aver osservato n voti positivi e m voti negativi, la distribuzione a *posteriori* è data da una beta PDF con $\alpha = n + 1$ e $\beta = m + 1$. A titolo di esempio, nella Figura 2.3 illustrata di seguito si possono vedere le PDF relative ad una distribuzione beta uniforme (Beta(p|1,1)) e ad una beta PDF a *posteriori* dopo aver osservato 7 voti positivi e 1 voto negativo (Beta(p|8,2)).



(a) Uniform PDF beta($p \mid 1, 1$) (b) PDF beta($p \mid 8, 2$)

Figura 2.3: Funzione di densità di probabilità beta.

Applicando questa teoria statistica ai sistemi di reputazione [14] si ottiene che la beta PDF rappresenta la probabilità con cui una prossima transazione con l'utente a cui fa riferimento la PDF considerata, abbia un esito positivo. Inoltre il punteggio di reputazione viene calcolato come valore atteso di questa PDF (nell'esempio fornito in precedenza si ha E(p) = 0.8), e questo può essere interpretato come la frequenza con cui le transazioni hanno esiti positivi: non si sa con certezza quante saranno le transazioni positive future, ma si può assumere che mediamente si avrà una percentuale di successo pari a E(p). Nell'esempio precedente, si ha che in media 8 transazioni su 10 avranno un esito positivo.

Il vantaggio dei sistemi Bayesiani consiste nel fatto che forniscono una solida base per il calcolo dei punteggi di reputazione, in quanto possono essere calcolati in modo univoco applicando delle espressioni matematiche; uno svantaggio invece risiede nel fatto che questo sistema potrebbe risultare difficile da comprendere per gli utenti che lo utilizzano.

2.3.4 Modelli Discreti

Questo sistema, anzichè basarsi su misure di tipo continuo, è definito su un dominio di tipo discreto. Basandosi sul modello formulato da Abdul-Rahman & Hailes (2000) [1] la reputazione di un certo utente x può assumere uno dei seguenti quattro valori:

- molto affidabile (very trustworthy),
- affidabile (trustworthy),
- inaffidabile (untrustworthy),
- molto inaffidabile (very untrustworthy).

Per calcolare la reputazione di questo utente si utilizzano delle tabelle di ricerca (look-up tables). Nel caso in cui un utente abbia avuto un'esperienza diretta con x, questa può essere utilizzata per determinare la reputazione di tale utente, assumendo che l'esperienza personale rifletta la reale affidabilità di x, mentre le raccomandazioni riguardanti x che differiscono dalla propria esperienza indicano se x è sovrastimato o sottostimato e vengono quindi utilizzate per aggiornare il suo punteggio di reputazione. Le raccomandazioni che provengono da un utente il quale tende a sovrastimare x (fornisce un voto di reputazione più alto di quello reale) verranno declassate, e vice versa.

Ad esempio considerando che una grande quantità di utenti ha assegnato un punteggio di reputazione pari ad *affidabile* all'utente x, se un altro utente y afferma che x è *molto inaffidabile*, l'opinione di y viene considerata poco rilevante rispetto a quelle degli altri utenti e quindi la reputazione di x rimane comunque *affidabile*.

Il vantaggio di questo tipo di modello è che risulta essere molto semplice da usare per gli utenti, i quali al termine di una transazione devono fornire un voto sull'affidabilità della controparte scegliendo tra un insieme finito di possibilità; il suo svantaggio però consiste nel fatto che le misure di tipo discreto non si prestano ad essere dei solidi principi di calcolo, poichè comportano l'utilizzo di altre tecniche, come le tabelle di ricerca, difficili da applicare a grandi quantità di dati.

2.3.5 Modelli di Belief

La teoria del belief è correlata con la teoria della probabilità, a differenza di quest'ultima però la somma di tutte le possibili probabilità relative ad un determinato evento non necessariamente sommano a 1, questa rimanente probabilità viene interpretata come *incertezza*.

Jøsang [11, 12] ha proposto un metrica basata sulla credenza/fiducia che prende il nome di *opinione*. Questa viene rappresentata con una tupla $\omega_x^A = (b, d, u, a)$ la quale esprime la credenza dell'utente A riguardo alla verità dell'evento x . I termini b, d e u rappresentano rispettivamente la credenza ($belief$), diffidenza ($disbelief$) e incertezza ($uncertainty$), con b, d, $u \in [0, 1]$ tali che $b + d + u = 1$. Invece il parametro $a \in$ [0,1] viene detto atomicità relativa, ed è il tasso di probabilità di base in assenza di prove. Tipicamente viene definito $a = 0.5$ poichè l'atomicità relativa viene interpretata come la probabilità in assenze di prove che l'utente con il quale viene intrapresa una transazione sia affidabile e cioè la probabilità che la transazione vada a buon fine.

Il valore atteso dell'opinione è calcolato come: $\mathrm{E}[\ \omega_x^A] = b + au$.

Se l'enunciato x viene definito come "l'utente Y è affidabile e onesto", possiamo applicare questo modello e interpretare l'*opinione* come l'affidabilità dell'utente Y.

In assenza di prove la probabilità che l'utente Y sia onesto, cioè la probabilità che una transazione vada a buon fine, è 0.5 e quindi sarà $a = 0.5$. Considerando ora un totale di 10 utenti e supponendo che tra questi 7 affermano che Y è affidabile mentre i rimanenti 3 lo ritengono inaffidabile allora l'utente Y sarà ritenuto affidabile al 70%. Se poi ci si fida solamente di 2 tra i 3 utenti che ritengono Y inaffidabile (su un totale di 10 utenti) allora Y sarà ritenuto inaffidabile al 20%. Sarà quindi $b = 0.7$ e $d = 0.2$. La rimanente probabilità, cioè 0.1, è l'incertezza: $u = 0.1$. In questo modo la tupla che rappresenta la credenza dell'utente A riguardo la verità dell'enunciato x è $\omega_x^A = (0.7, 0.2, 0.1, 0.5)$. Il valore atteso dell'opinione di A riguardo all'affidabilità di Y risulta quindi essere $\mathrm{E}[\ \omega_x^A] = b + au = 0.7 + 0.5 \cdot 0.1 = 0.75$ e può essere utilizzato come punteggio di reputazione per Y.

Anche Yu & Singh (2002) [25] hanno proposto un modello che calcola i punteggi di reputazione basandosi su queste teorie. In questo modello, considerato un utente A, questo può essere affidabile (T_A) o non affidabile ($\neg T_A$) e vengono considerate le credenze relative all'affidabilità di A, $m(T_A)$, e alla sua inaffidabilità, $m(\neg T_A)$. A questo punto si può definire il punteggio di reputazione Γ dell'utente A come:

$$\Gamma(A) = m(T_A) - m(\neg T_A), \tag{2.5}$$

dove $m(T_A)$, $m(\neg T_A) \in$ [0, 1] e $\Gamma(A) \in$ [-1, 1] . $\Gamma(A) = -1$ significa che l'utente A è ritenuto completamente inaffidabile, mentre $\Gamma(A) = 1$ completamente affidabile.

I punteggi forniti dai singoli utenti sono delle misure di credibilità, che possono essere affidabile (T_A) o non affidabile ($\neg T_A$). Queste singole misure vengono poi combinate per ottenere i valori della credibilità $m(T_A)$ e $m(\neg T_A)$, e vengono usate nell'equazione (2.5) per calcolare il puteggio di reputazione.

Riprendendo l'esempio fornito in precedenza, se su un totale di 10 utenti 7 di questi reputano A affidabile (T_A), e i rimanenti 3 lo reputano inaffidabile ($\neg T_A$), ma ci si fida solamente di 2 tra questi 3, la credenza riguardante l'affidabilità di A sarà $m(T_A) = 0.7$, mentre quella relativa alla sua inaffidabilità sarà $m(\neg T_A) = 0.2$. Si può quindi calcolare il puteggio di reputazione per l'utente A come $\Gamma(A) = m(T_A) - m(\neg T_A) = 0.7 - 0.2 = 0.5$.

2.3.6 Modelli di Flusso

I *modelli di flusso* (flow models) sono dei sistemi che calcolano la reputazione in modo iterativo, tramite delle catene arbitrariamente lunghe.

Alcuni di questi modelli considerano un peso di reputazione costante per tutta la comunità formata dagli utenti appartenenti al sistema (anche se questa non è una condizione necessaria), e lo distribuiscono tra i vari membri. Vale a dire che la somma totale dei pesi associati a tutti gli utenti rimane costante, e se un utente aumenta la sua reputazione allora quella di tutti gli altri utenti viene diminuita. Inoltre la reputazione di un utente aumenta in funzione dei flussi in ingresso, e diminuisce in funzione dei flussi in uscita.

Un esempio di un modello di questo tipo è Google PageRank [17]: un algoritmo di analisi che assegna un peso numerico ad ogni elemento di un *collegamento ipertestuale* (*hyperlink*) in un insieme di documenti, ad esempio pagine web, allo scopo di determinare la sua importanza all'interno di questo insieme. Questo peso numerico viene detto *PageRank* e viene calcolato in base al numero di *hyperlinks* che puntano alla pagina presa in considerazione. Il *PageRank* può essere visto come il punteggio di reputazione di una pagina, e ogni singolo *hyperlink* rappresenta un voto positivo.

La definizione che Page ha fornito per il *PageRank* è la seguente:

Sia u una pagina web appartenente all'insieme P (un insieme di pagine web con degli hyperlinks), e sia P_u un sottoisieme di P contenente le n pagine che hanno almeno un *hyperlink* che punta a u. Siano p_k gli elementi di P_u, e sia $E(u)$ un vettore su P corrispondente ad un rank di base per la pagina u. Allora il *PageRank* della pagina u sarà $PR[u]$

dato da:

$$PR[u] = cE(u) + c\sum_{k=1}^{n} \frac{PR[p_k]}{C(p_k)} \tag{2.6}$$

dove:

- PR[p_k] è il valore del PageRank della pagina p_k,
- C(p_k) è il numero complessivo di link contenuti nella pagina p_k,
- c è scelto in modo che $\sum_{u \in P} PR[u] = 1$,
- E(u) è un termine deciso da Google, con il quale si può decidere il valore minimo di PageRank che può avere una pagina e la percentuale di PageRank che viene distribuita tra le varie pagine (è preferibile scegliere E tale che $\sum_{u \in P} E(u) = 0.15$) [17].

Si può notare quindi che, all'aumentare dei link che puntano alla pagina u, il suo PageRank aumenta. Infatti, mentre il primo termine dell'equazione fornisce un valore basato su un rank iniziale, di base, il secondo termine rappresenta il valore del rank in funzione degli *hyperlinks* che puntano a u.

In base alla definizione fornita sopra il PageRank è un valore compreso nell'intervallo [0, 1], ma in realtà nel sistema di PageRank utilizzato da Google il punteggio è scalato all'interno dell'intervallo [0, 10], e vengono presi in considerazione anche altri elementi, con lo scopo di rendere difficile l'alterazione del PageRank.

In conclusione il PageRank può essere visto come un sistema di reputazione in cui il punteggio di reputazione è rappresentato proprio dal PageRank, gli hyperlink corrispondono ai voti positivi, mentre non sono presenti i voti negativi.

Nella Tabella 2.1 fornita nella pagina successiva si vede una possibile classificazione dei sistemi di reputazione in base al metodo utilizzato per il calcolo dei punteggi di reputazione e vengono brevemente illustrati quali sono i loro pro e contro.

2.4 Sistemi di Reputazione nella Computer Security

Nel contesto odierno un problema che richiede sempre più attenzione è quello della computer security (detta anche IT security). Lo scopo è

	PRO	CONTRO
Somma semplice dei voti (eBay)	- semplice da usare e comprendere	- scarsa indicazione della reputazione - voti fraudolenti
Media dei voti (Amazon)	- semplice da usare e comprendere	- voti fraudolenti
Sistemi Bayesiani	- solida base per il calcolo dei punteggi	- difficile da comprendere per gli utenti
Sistemi discreti	- semplice da usare per gli utenti	- non solido principio di calcolo
Modelli di belief	- semplice da usare per gli utenti	- difficile da implementare su una comunità molto numerosa
Modelli di flusso (PageRank)	- indicazione precisa della reputazione	- difficile da implementare su una comunità molto numerosa

Tabella 2.1: classificazione dei sistemi di reputazione.

quello di garantire agli utenti una protezione per le loro apparecchiature, informazioni o servizi contro altri utenti maligni. Un modo per proteggere le informazioni è limitare l'accesso alle risorse solamente agli utenti autorizzati (access control). In alcuni casi però il problema è contrario: si deve proteggere l'utente da chi offre delle risorse maligne (software dannosi, come virus). In questo caso i tradizionali sistemi di sicurezza non sono adatti, poichè ad esempio non sono in grado di determinare se un utente fornisce informazioni false. In questo contesto si possono introdurre i sistemi di reputazione: condividendo informazioni riguardanti gli utenti del sistema si possono distinguere quali utenti sono affidabili e quali invece agiscono in modo disonesto.

Si possono quindi distinguere due tipi di sicurezza: *hard security* che indica le tecniche tradizionali di sicurezza (come l'accesso controllato), e *soft security* che invece è formata da quei meccanismi di controllo che si basano sulla condivisione di informazioni da parte di una comunità. I sistemi di reputazione vanno perciò inseriti nell'ambito della soft security [15].

Ultimamente i sistemi di reputazione hanno iniziato ad avere un più largo impiego in questo campo, con l'effetto di introdurre il concetto di affidabilità e attendibilità come misura di sicurezza, anche se per determinare tale misura tipicamente si utilizza il concetto di *security assurance level*

[15]. Questo concetto può esere interpretato come la resistenza che un sistema ha nei confronti di attacchi maligni, o più informalmente come il livello di affidabilità che la comunità gli attribuisce.

Un altro ambito nel quale la sicurezza ha un ruolo importante è quello delle comunicazioni: vengono utilizzate delle tecniche di crittografia sui canali di comunicazione e sull' autenticazione dell' identità degli utenti per garantire la confidenzialità dei dati. Per descrivere la *Certification Autority* (CA) che fornisce i servizi usati per determinare le identità dei membri, si utilizza il termine *"trust provider"*, dimostrando così l'influenza che i sistemi di reputazione hanno nel settore. Dal momento che gli utenti sono anche interessati a conoscere l'affidabilità del resto degli utenti o la qualità dei servizi, si devono introdurre dei sistemi di reputazione con i quali si può calcolare la cosiddetta *provision trust*, cioè la fiducia che un utente ha in un servizio o una risorsa fornita da un altro utente.

Capitolo 3

Sistemi di Raccomandazione

I sistemi di raccomandazione hanno trovato applicazione principalmente nei siti di E-commerce, dove vengono utilizzati per raccomandare agli utenti oggetti che non hanno ancora visto. Questo avviene facendo una stima del voto che l'utente potrebbe dare ad un determinato oggetto, basandosi principalmente sui precedenti voti che l'utente ha fornito per altri oggetti simili e sulle sue preferenze. Una volta calcolate queste stime, si possono raccomandare all'utente gli oggetti che hanno ottenuto dei punteggi (stimati) più alti.

Formalmente: sia C un insieme di utenti, e sia S l'insieme dei possibili oggetti che possono essere raccomandati. Sia u una funzione di utilità, che misura quanto un oggetto s può essere utile all'utente u, cioè una funzione del tipo: $u : C \times S \longrightarrow \mathbb{R}$. Allora il problema di un sistema di raccomandazione è quello di scegliere per ogni utente $c \in C$ l'oggetto $s' \in S$ tale che massimizza la funzione di utilità per l'utente u:

$$\forall c \in C, \qquad s' = \arg\max_{s \in S} u(c, s) \tag{3.1}$$

La funzione di utilità può essere una funzione arbitraria, che dipende anche dall' applicazione nella quale deve essere implementata, ma tipicamente viene rappresentata dai voti degli oggetti, e indica quanto ad un particolare utente è piaciuto un determinato oggetto. Nel caso più semplice, gli elementi dello spazio C (gli utenti), così come quelli dello spazio S (gli oggetti), sono definiti da un solo parametro, l'ID usato per identificare l'elemento; ma in realtà ad ogni elemento si possono associare molti parametri. Ad esempio nel caso di un utente si può creare un profilo che contiene informazioni riguardanti l'età, il genere e altri elementi che poi si possono usare per formulare le raccomandazioni, come per esempio il luogo in cui vive l'utente o le sue preferenze. Il problema maggiore dei sistemi di raccomadazione è che la funzione di utilità non è

definita in tutto lo spazio $C \times S$, ma solo in un suo sottoinsieme, formato dalle coppie di utenti e oggetti in cui i primi hanno fornito un voto per i secondi. Si tratta quindi di dover stimare i voti per tutti gli altri oggetti non ancora votati dall'utente.

In base all'approccio utilizzato per stimare questi voti, i sistemi di raccomandazione possono essere classificati in queste tre categorie:

- *Raccomandazioni basate sul contenuto* (Content-based Recommendations): raccomandano all'utente gli oggetti simili a quelli che ha già votato e preferito in passato,
- *Raccomandazioni collaborative* (Collaborative Recommendations): raccomandano all'utente gli oggetti che altri utenti con gusti simili hanno votato in passato,
- *Approcci ibridi* (Hybrid Approaches): sistemi che combinano i due metodi sopra citati.

In aggiunta, oltre a questi sistemi di raccomandazione in cui vengono predetti in modo *assoluto* i valori dei voti che un individuo darebbe agli oggetti non ancora visti, ci sono altri sistemi che formulano le predizioni per le preferenze *relative* dell'utente, si parla di *filtraggio basato sulle preferenze*. In questo tipo di sistemi nel formulare una raccomandazione invece di considerare tutti gli oggetti appartenenti al sistema, si considerano solamente gli oggetti non ancora visti dall'utente ma che sono in linea con le sue preferenze.

3.1 Metodi Basati sul Contenuto

Nei metodi di questo tipo, la funzione di utilità $u(c, s)$ dell'oggetto s per l'utente c viene calcolata basandosi sui valori di utilità $u(c, s_i)$ forniti dallo stesso utente c per altri oggetti $s_i \in S$ simili a s.

Principalmente questi metodi vengono utilizzati in sistemi in cui gli oggetti da raccomandare contengono informazioni di tipo testuale, come documenti o pagine web. Nel caso di un documento (o di qualsiasi altro elemento di tipo testuale), si può definire *Content(s)* come il profilo del documento s, costituito da un insieme di parole chiave di s. L'importanza di una parola k_i in un documento d_j si può calcolare con delle misure di peso w_{ij}, basandosi sui metodi dell' *Information Retrieval* come il TF-IDF (*term frequency/inverse document frequency*). Assumendo che N sia il numero totale di documenti che possono essere raccomandati ad un utente, e che la parola k_i compaia in n_i di questi, con $f_{i,j}$ uguale al

numero di volte che la parola k_i compare nel documento d_j, allora la *term frequency* $TF_{i,j}$ della parola k_i nel documento d_j è definita come segue:

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}} \tag{3.2}$$

dove $max_z f_{z,j}$ rappresenta la massima frequenza tra tutte le parole k_z che appaiono nel documento d_j.

Invece la misura della *inverse document frequency* (IDF_i) della parola k_i è definita come:

$$IDF_i = \log \frac{N}{n_i} \tag{3.3}$$

A questo punto si può definire il peso $w_{i,j}$ TF-IDF per la parola k_i nel documento d_j, e il contenuto del documento d_j, $Content(d_j)$ come:

$$w_{i,j} = TF_{i,j} \times IDF_i \tag{3.4}$$

$$Content(d_j) = (w_{1j}, ...w_{kj}) \tag{3.5}$$

Considerato il profilo dell'utente c, contenente i suoi gusti e preferenze, questo può essere definito come un vettore di pesi $ContentBasedProfile(c) = (w_{1c}, ...w_{kc})$, dove ogni peso w_{xc} rappresenta l'importanza della parola chiave k_i per l'utente c, e si può calcolare basandosi sulle preferenze e i voti forniti direttamente dall'utente. Si può quindi definire generalmete la funzione di utilità $u(c, s)$ come:

$$u(c, s) = score(ContentBasedProfile(c), Content(s)). \tag{3.6}$$

I due profili, dell'utente u e dell'oggetto s, possono essere rappresentati da dei vettori, rispettivamente $\vec{w_c}$ e $\vec{w_s}$. Quindi la funzione di utilità può essere calcolata proprio in funzione di questi due vettori, ad esempio utilizzando la *misura di somiglianza del coseno*:

$$u(c, s) = \cos(\vec{w_c}, \vec{w_s}) = \frac{\vec{w_c} \cdot \vec{w_s}}{||\vec{w_c}||_2 \times ||\vec{w_s}||_2} = \frac{\sum_{i=1}^{k} w_{i,c} w_{i,s}}{\sqrt{\sum_{i=1}^{k} w_{i,c}^2} \sqrt{\sum_{i=1}^{k} w_{i,s}^2}} \tag{3.7}$$

dove k è il numero totale di parole chiave presenti nel sistema.

Ad esempio se l'utente c preferisce leggere articoli riguardanti la fotografia, un sistema di raccomandazione del tipo *content-based* sarà in grado di fornire a questo utente altri articoli pertinenti a questo campo. Infatti, specificando nel profilo dell'utente la preferenza per il settore

della fotografia, tutte le parole presenti in un articolo che si riferiscono a questo ambito (ad esempio termini tecnici) verranno pesate con un valore superiore rispetto alle altre parole. Quindi usando la misura di somiglianza del coseno, a questi articoli verrà attribuita una misura di utilità maggiore, poichè questi conterranno molte parole che hanno un peso elevato, mentre gli articoli relativi ad altri argomenti avranno un'utilità inferiore.

Se ad esempio si considera il documento j che contiene 100 parole e nel quale la parola *fotografia* (i) compare 5 volte, il fattore $TF_{i,j}$ per tale parola è $TF_{i,j} = \frac{5}{100} = 0.05$. Considerando ora una collezione di 1000 documenti all'interno della quale la parola *fotografia* compare in 10 di questi, il termine IDF_i sempre per tale parola è $IDF_i = \log \frac{1000}{10} = 2$. A questo punto si può calcolare il termine TF-IDF come $w_{i,j} = TF_{i,j} \times IDF_i = 0.05 \times 2 = 0.1$.

Per quanto riguarda la misura di somiglianza del coseno, per semplicità, ci si può limitare al caso bidimensionale, ad esempio considerando solamente l'importanza che possono avere le due parole *video* e *fotografia*. Supponendo che i profili dell'utente c e dell'oggetto s siano rispettivamente $\vec{w}_c = (0.6, 0.8)$ e $\vec{w}_s = (0.8, 0.7)$, (dove gli elementi di $\vec{w}_c$ rappresentano l'importanza delle due parole per l'utente c e quelli di $\vec{w}_s$ l'importanza delle parole all'interno dell'oggetto s), applicando la formula (3.7) risulta essere circa $u(c, s) = 0.9$. In base alla definizione di $u(c, s)$ fornita in (3.7), dati due vettori se si ottiene un valore pari a 1 allora i due profili (utente e oggetto) hanno la massima somiglianza, e quindi quasi certamente se si raccomanda l'oggetto s a c l'utente lo apprezzerà.

Esistono anche altri metodi di calcolo dell'utilità, tra i quali i classificatori Bayesiani [16] e varie tecniche nell'ambito dell'*apprendimento automatico (machine learning)* (come alberi di decisione e reti neurali artificiali) [19]. Queste tecniche differiscono dalla misura di somiglianza del coseno in quanto anzichè calcolare l'utilità tramite una formula euristica, si basano sull'apprendimento di un *modello* ottenuto con l'osservazione di dati.

I sistemi di raccomandazioni content-based hanno però alcune limitazioni: in primo luogo questi sistemi sono limitati dalle caratteristiche degli oggetti che vengono considerate nella formulazione delle raccomandazioni. Ad esempio un sistema che raccomanda articoli da leggere non può essere usato direttamente per raccomandare materiale elettronico, ma deve essere modificato e adattato. Bisogna quindi fare in modo che il contenuto degli elementi del sistema sia formulato in modo da poter essere convertito o adattato automaticamente, altrimenti bisogna asse-

gnare manualmente le caratteristiche degli oggetti. Inoltre se due oggetti diversi vengono rappresentati allo stesso modo (con lo stesso insieme di attributi) questi risultano essere indistinguibili. Un'altra limitazione consiste nella iperspecializzazione, cioè il sistema è in grado di raccomandare solamente oggetti che si abbinano con il profilo dell'utente (in termini di misura di utilità) e quindi all'utente verranno raccomandati solo oggetti simili a quelli che lui ha già votato. Il problema maggiore però è rappresentato dai nuovi utenti: il sistema è in grado di fornire raccomandazioni accurate solo se l'utente ha votato un sufficiente numero di oggetti, in modo da riuscire a determinare le sue preferenze. Un metodo per risolvere questo problema consiste nel far votare ai nuovi utenti una lista di oggetti predefiniti, in modo da ottenere le informazioni necessarie riguardanti quegli utenti prima che il sistemi inizi a fornire le varie raccomandazioni.

3.2 Metodi Collaborativi

A differenza dei sistemi di raccomandazione "content-based", i sistemi collaborativi, detti anche sistemi di *filtraggio collaborativo*, stimano il valore della funzione di utilità di un particolare oggetto per un determinato utente basandosi sugli oggetti simili precedentemente votati dagli *altri* utenti con gusti simili a lui. Detta $u(c, s)$ l'utilità dell'oggetto s per l'utente c, questa viene stimata combinando i valori dell'utilità $u(c_i, s)$ assegnati all'oggetto s da un insieme di utenti $c_i \in C$ che risultano essere *simili* a c, cioè utenti che hanno dei gusti simili a c.

Secondo Balabanovic e Shoham [6] gli algoritmi dei sistemi di raccomandazione collaborativi possono essere divisi in due classi:

- algoritmi basati sulla memoria (o sull'euristica),
- algoritmi basati sui modelli.

3.2.1 Algoritmi Basati sulla Memoria

Negli algoritmi basati sulla memoria (memory-based) le predizioni vengono fatte basandosi su una collezione di oggetti precedentemente votati dagli utenti. Il valore di un voto sconosciuto $r_{c,s}$ di un oggetto s per l'utente c viene calcolato come un aggregato dei voti di altri utenti (ad esempio i più simili a c) forniti per lo stesso oggetto s:

$$r_{c,s} = aggr_{c' \in \hat{C}} r_{c',s} \tag{3.8}$$

dove $\hat{C}$ è l'insieme degli N utenti più simili a c, i quali hanno già votato l'oggetto s. Alcuni esempi della funzione di aggregazione $r_{c,s}$ sono i seguenti:

- $r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s}$
- $r_{c,s} = k \sum_{c' \in \hat{C}} sim(c, c') \times r_{c',s}$
- $r_{c,s} = \bar{r_c} + k \sum_{c' \in \hat{C}} sim(c, c') \times (r_{c',s} - \bar{r}_{c'})$

dove k è un fattore di normalizzazione, tipicamente viene scelto come $k = 1/\sum_{c' \in \hat{C}} |sim(c, c')|$, e $\bar{r}_c$ è il voto medio fornito dall'utente c, definito come:

$$\bar{r}_c = \frac{1}{|S_c|} \sum_{s \in S_c} r_{c,s} \text{ , con } S_c = \{s \in S | r_{c,s} \neq \emptyset\}. \tag{3.9}$$

Il primo esempio è il caso più semplice, dove la funzione di aggregazione è calcolata come media di tutti i voti forniti dagli utenti $c' \in \hat{C}$, cioè quelli più simili a c. Nei secondi due casi viene utilizzata una media pesata dei voti, nella quale i voti vengono pesati con la funzione di somiglianza $sim(c, c')$, che tipicamente è una misura di distanza. Più i due utenti sono simili maggiore sarà il peso attribuito al voto $r_{c',s}$ nella predizione di $r_{c,s}$. Questa funzione viene usata per differenziare vari livelli di somiglianza.

Per calcolare la somiglianza ci sono vari approcci, tutti comunque basati sui voti degli oggetti votati da entrambi gli utenti. Gli approcci più usati sono la *correlazione* e il *coseno di somiglianza*. Definendo $S_{x,y}$ come l'insieme degli oggetti che entrambi gli utenti x e y hanno votato, cioè $S_{x,y} = \{s \in S | \quad r_{x,s} \neq \emptyset \quad \& \quad r_{y,s} = \emptyset\}$, si può definire la misura di correlazione come:

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r_x})(r_{y,s} - \bar{r_y})}{\sqrt{\sum_{s \in S_{x,y}} (r_{x,s} - \bar{r_x})^2 (r_{y,s} - \bar{r_y})^2}} \tag{3.10}$$

Il coseno di somiglianza invece viene definito come nel caso della funzione di utilità in un sistema di raccomandazione *content-based*, considerando gli utenti come dei vettori $\vec{x}$ e $\vec{y}$ all'interno di uno spazio vettoriale di dimensione m, con $m = |S_{xy}|$:

$$sim(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}||_2 \times ||\vec{y}||_2} = \frac{\sum_{s \in S_{xy}} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2} \sqrt{\sum_{s \in S_{xy}} r_{y,s}^2}} \tag{3.11}$$

Ad esempio considerando l'utente x il quale non ha ancora votato l'oggetto s, si può stimare il voto che presumibilmente lui darebbe a questo oggetto usando la media dei voti forniti per l'oggetto s dagli N utenti

più simili a x. Assumendo che il voto di utilità che gli utenti possono fornire sia un numero nel range da 0 a 1 e che i quattro utenti più simili ad x (cioè gli utenti $c' \in \hat{C}$) abbiano fornito i seguenti voti 0.6, 0.8, 0.7 e 0.9 allora la stima risultante è $r_{x,s} = \frac{1}{4} \sum_{c' \in \hat{C}} r_{x',s} = 0.75$. Inoltre supponendo che l'utente x abbia votato un totale di cinque oggetti ($s \in S_x$) assegnando i seguenti voti 0.4, 0.9, 0.8, 0.8, e 0.7 allora si può calcolare il voto medio fornito da x come $\bar{r}_x = \frac{1}{5} \sum_{s \in S_x} r_{x,s} = 0.72$. Se invece si vuole calcolare la somiglianza tra l'utente x sopra descritto e un utente y che ha votato gli stessi oggetti di x con i seguenti punteggi 0.6, 0.9, 0.7, 0.8 e 0.7, quindi con punteggio medio $\bar{r}_y = 0.76$, si può usare il coseno di somiglianza descritto nell'equazione (3.11) ottenendo il seguente risultato: $sim(x, y) = 0.98$.

La differenza tra la misura del coseno di somiglianza usato nel caso di un sistema di raccomandazione content-based e in uno di tipo collaborativo consiste nel fatto che nel primo caso viene usato per misurare la somiglianza tra vettori di pesi TF-IDF, mentre nel secondo misura la somiglianza tra vettori di voti forniti dagli utenti.

Esistono anche altri approcci per il calcolo della somiglianza [22], e solitamente ogni sistema di raccomandazione usa degli approcci diversi, in modo da calcolare la somiglianza nel modo più efficiente possibile. Tipicamente si calcolano i valori di somiglianza di tutti gli utenti e poi vengono ricalcolati di tanto in tanto, ad esempio quando un utente richiede questi valori, poichè le caratteristiche degli utenti non cambieranno drasticamente in un breve periodo.

Queste stesse tecniche possono essere utilizzate anche per calcolare la somiglianza tra gli oggetti, anzichè tra gli utenti.

3.2.2 Algoritmi Basati sui Modelli

Gli algoritmi basati sui modelli invece utilizzano le collezioni di voti per apprendere un *modello* usato poi per predire i voti non conosciuti.

Ad esempio [7] ha proposto un approccio di tipo probabilistico al filtraggio collaborativo, dove i voti sconosciuti vengono stimati come valore atteso:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times Prob(r_{c,s} = i \quad | \quad r_{c,s'}, s' \in S_c) \tag{3.12}$$

assumendo che i valori che i voti possono assumere siano dei numeri interi compresi in [0, n] e la funzione di probabilità esprima la probabilità che l'utente c attribuisca il voto i al particolare oggetto s dati i valori dei

precedenti voti forniti da c per gli oggetti s' (gli oggetti simili a s). Questa probabilità può essere stimata in diversi modi tra i quali i più importanti sono :

- modelli a gruppi (*cluster models*)
- reti Bayesiane

Nel primo modello gli utenti vengono raggruppati in classi, mentre nel secondo vengono rappresentati come nodi all'interno di una rete Bayesiana [3]. In entrambi i casi le strutture vengono apprese con delle tecniche di *machine learning* osservando una certa quantità di dati.

Quindi la differenza tra i sistemi basati sulla memoria e quelli basati sui modelli è che i primi formulano le predizioni calcolando l'utilità degli oggetti tramite formule euristiche, mentre i secondi si basano su un modello appreso osservando i dati collezionati.

3.3 Problematiche dei Sistemi Content-based e Collaborativi

I problemi principali che un sistema di raccomandazione deve affrontare sono:

- problema dei nuovi utenti,
- problema dei nuovi oggetti,
- scarsità dei voti.

Il primo è un problema che affligge sia i sistemi collaborativi sia i sistemi basati sul contenuto: prima di fornire raccomandazioni affidabili a un utente il sistema deve collezionare un numero sufficiente di voti, in modo da apprendere le preferenze degli utenti. Dal momento che un sistema collaborativo usa unicamente le preferenze degli utenti, quando viene aggiunto un nuovo oggetto, prima che questo possa essere raccomandato deve essere votato da altri utenti. Il numero di voti che vengono collezionati dal sistema solitamente è molto minore rispetto a quello dei voti che devono essere predetti. Inoltre ci possono essere molti oggetti che vengono votati solo da pochi utenti, ad esempio in un sistema di raccomandazione di libri, ci saranno alcuni libri che vengono votati raramente. Per di più per gli utenti che hanno dei gusti particolari, insoliti rispetto a quelli degli altri utenti, non ci saranno altri membri che risultano essere

simili a questi, e perciò le raccomandazioni per questi utenti risulteranno essere poco accurate. Per cercare di risolvere questo problema si possono prendere in considerazione le informazioni fornite dal profilo degli utenti. In questo modo due utenti saranno simili non solo se hanno gli stessi gusti, ma anche se i due profili si assomigliano, ad esempio hanno circa la stessa età, vivono in aree vicine etc. Questa estensione dei sistemi collaborativi tradizionali prende il nome di *demographic filtering* [18].

Invece le prime due problematiche che riguardano i nuovi utenti e oggetti possono essere risolte utilizzando dei *metodi ibridi* che utilizzano contemporaneamente dei metodi collaborativi e basati sul contenuto. Questi metodi vengono descritti di seguito.

3.4 Metodi Ibridi

I sistemi di raccomandazione ibridi combinano i metodi collaborativi e quelli basati sul contenuto per cercare di evitare alcune problematiche caratteristiche di questi, come il problema dei nuovi utenti per quelli basati sul contenuto o quello dei nuovi oggetti per i sistemi collaborativi. I metodi collaborativi e basati sul contenuto possono essere combinati in modi differenti:

1. implementando i metodi collaborativi e basati sul contenuto separatamente per poi combinare le loro predizioni,

2. includendo alcune caratteristiche dei metodi basati sul contenuto all'interno di un sistema di tipo collaborativo,

3. includendo alcune caratteristiche dei metodi collaborativi all'interno di un sistema basato sul contenuto,

4. costruire un modello unificato che fonde le caratteristiche sia dei modelli collaborativi sia di quelli basati sul contenuto.

Nel primo caso ci sono due possibili soluzioni: combinare i risultati, cioè i voti predetti, ottenuti dai singoli sistemi, in un unico risultato finale, ad esempio effettuando una combinazione lineare dei due voti ottenuti, oppure scegliere di usare individualmente uno dei due sistemi implementati, scegliendo il sistema che in quella situazione è in grado di fornire la migliore raccomandazione, basandosi su delle metriche di *qualità* delle raccomandazioni [3].

Se ad esempio con il sistema collaborativo si ottiene una stima per il voto che l'utente c darebbe all'oggetto s pari a $r^1_{c,s} = 0.85$, mentre con

il sistema content-based si ottiene $r^2_{c,s} = 0.79$, allora si può pensare di combinare i due voti effettuando una media tra questi, ottenendo una stima totale pari a $r_{c,s} = \frac{1}{2}(0.85 + 0.79) = 0.82$.

Nel secondo caso si ottiene un sistema con le tradizionali tecniche dei sistemi collaborativi che però mantiene i profili degli utenti descritti per i sistemi basati sul contenuto. In questo modo per calcolare la somiglianza tra due utenti vengono usati non solo gli oggetti che sono stati votati da entrambi, ma anche i loro profili. Così facendo si riesce ad evitare il problema della scarsità dei voti, inoltre all'utente possono essere raccomandati non solo gli oggetti altamente apprezzati dagli utenti simili, ma anche quegli oggetti che si abbinano al profilo dell'utente.

Nel terzo caso l'approccio più usato è quello di far ricorso a delle tecniche di *riduzione della dimensionalità* su un gruppo di profili content-based. Ad esempio partendo da una collezione di profili di utenti (che contengono molte caratteristiche e sono perciò multidimensionali) questi vengono rappresentati come dei vettori di dimensione inferiore sui quali poi si applicano le tecniche tipiche di un sistema content-based [23].

L'ultimo caso invece unifica le varie caratteristiche dei due metodi in un modello generale. Un esempio di questi sistemi ibridi è stato proposto da [5], il quale utilizza le informazioni degli utenti e degli oggetti in un unico modello statistico che stima i voti sconosciuti r_{ij} per l'utente i e l'oggetto j:

$$r_{ij} = x_{ij}\mu + z_i\gamma_j + w_j\lambda_i + e_{ij}, \tag{3.13}$$

$$e_{ij} \sim N(0, \sigma^2), \tag{3.14}$$

$$\lambda_i \sim N(0, \Lambda), \tag{3.15}$$

$$\gamma_j \sim N(0, \Gamma), \tag{3.16}$$

dove $i = 1, ..., I$ e $j = 1, ..., J$ sono rispettivamente gli utenti e gli oggetti. e_{ij}, λ_i e γ_j sono delle variabili aleatorie che rappresentano il rumore, sorgenti di eterogeneità degli utenti ed eterogeneità degli oggetti. x_{ij} è una matrice che contiene le caratteristiche di utenti e oggetti, z_i è un vettore di caratteristiche dell'utente, w_j è un vettore di caratteristiche dell'oggetto [3]. I parametri incogniti in questo modello sono μ, σ^2, Λ e Γ i quali vengono stimati tramite i voti già conosciuti. Quindi si utilizzano gli attributi dell'utente $\{z_i\}$ (che fanno parte del profilo dell'utente), gli attributi dell'oggetto $\{w_j\}$ (che fanno parte del profilo dell'oggetto), e le loro interazioni per stimare il voto sconosciuto di un oggetto.

In conclusione i sistemi di raccomandazione vengono classificati in base all'approccio usato per la raccomandazione in *sistemi basati sul contenuto, collaborativi* o *ibridi* e in base alle tecniche utilizzate per calcolare

	Basati sull'euristica	Basati sui Modelli
Content-Based	Tecniche usate: - TF/IDF (term frequency inverse document frequency)	Tecniche usate: - classificatori bayesiani - alberi di decisione - reti neurali artificiali
Collaborativi	Tecniche usate: - utenti più simili - coseno di somiglianza - correlazione - teoria dei grafi	Tecniche usate: - reti bayesiane - reti neurali artificiali - modelli probabilistici
Ibridi	Tecniche usate: - combinazione lineare dei voti - incorporare una componente dei sistemi basati sui modelli in un sistema euristico	Tecniche usate: - costruire un modello unificato - incorporare una componente dei sistemi euristici in un sistema basato sui modelli

Tabella 3.1: classificazione dei sistemi di raccomandazione.

la stima dei voti sconosciuti in *sistemi basati sull'euristica* o *basati sui modelli*. La Tabella 3.1 mostra in che modo i sistemi di raccomandazione possono essere raggruppati.

3.4.1 Estendere i Sistemi di Raccomandazione

Entrambi i sistemi collaborativi e content-based soffrono di alcune limitazioni (quelle descritte in precedenza) e quindi, per fornire delle raccomandazioni migliori e più affidabili o anche per usare dei sistemi di raccomandazione in applicazioni più complesse, è necessario estendere i sistemi sopra descritti. Per migliorare questi sistemi si possono ad esempio incorporare delle informazioni contestuali all'interno del processo di raccomandazione e fornire raccomandazioni più flessibili e meno invasive per l'utente.

La maggior parte dei metodi di raccomandazione forniscono dei voti basandosi su conoscenze limitate riguardanti l'utente e l'oggetto senza prendere in considerazione le informazioni sulla cronologia delle transazioni dell'utente. Ad esempio i metodi di filtraggio collaborativo non prendono nemmeno in considerazione i profili dell'utente e dell'oggetto, ma considerano solamente le informazioni relative ai voti. Quindi una strategia per migliorare questi sistemi consiste nell'introdurre nei profili delle informazioni più avanzate sia per gli utenti che per gli oggetti, come

per esempio le preferenze dell'utente. Dopo aver creato i profili, si può definire una generale funzione di stima dei voti nel modo seguente [3]:

Sia $\vec{c_i} = (a_{i1}, ..., a_{ip})$ un vettore di dimensione p che rappresenta il profilo dell'utente i, e sia $\vec{s_j} = (b_{j1}, ..., b_{jr})$ un vettore che contiene le r caratteristiche dell'oggetto j. Gli elementi a_{ik} e b_{jl} possono essere dei concetti diversi in base all'applicazione che si deve implementare, ad esempio possono essere numeri, categorie, regole etc. Sia $\vec{c}$ un vettore contenente tutti gli utenti, $\vec{c} = (\vec{c_1}, ..., \vec{c_m})$, e sia $\vec{s} = (\vec{s_1}, ..., \vec{s_n})$ il vettore contente tutti gli oggetti. Allora la funzione più generale che stima i voti sconosciuti per l'utente i e l'oggetto j è:

$$r'_{ij} = \begin{cases} r_{ij}, & \text{se } r_{ij} \neq \emptyset; \\ u_{ij}(R, \vec{c}, \vec{s}), & \text{se } r_{ij} = 0. \end{cases}$$

dove $R = \{r_{ij} \neq \emptyset\}$ è una matrice che contiene tutti i voti conosciuti, e u_{ij} è la funzione di utilità.

Questa rappresenta la definizione più generale dove la stima dipende non solo dai voti conosciuti forniti da tutti gli altri utenti (R), ma anche dal profilo dell'utente $\vec{c_i}$ e da tutti gli altri utenti ($\vec{c}$), dal profilo dell'oggetto $\vec{s_j}$ e possibilmente anche da tutti gli altri oggetti ($\vec{s}$). Tipicamente i sistemi di raccomandazione usano una funzione di utilità u_{ij} che dipende solo da un piccolo sottoinsieme dello spazio che contiene gli input R, $\vec{c}$, $\vec{s}$. Ad esempio i metodi di filtraggio collaborativo basati sulla memoria utilizzano solamente la colonna R_j della matrice R, tipicamente limitandosi esclusivamente agli N utenti più simili a $\vec{c_i}$, e non cosiderano $\vec{c}$ ed $\vec{s}$.

Un altro modo per migliorare i sistemi di raccomandazione è quello di ricorrere alla *teoria dell'approssimazione matematica*, ad esempio utilizzando delle *funzioni di base radiale* per definire la funzione di utilità [8]. Una funzione di base radiale può essere definita come segue:

Dato un insieme di punti $X = \{x_1, ..., x_m\}$ (con $x_i \in \mathbb{R}^N$) e i valori che una funzione sconosciuta f assume in questi punti, $f(x_1)$, ..., $f(x_m)$, una funzione di base radiale $r_{f,X}$ stima i valori che la funzione f assume in tutto $\mathbb{R}^N$, assumendo che $r_{f,X}(x_i) = f(x_i)$ per ogni $i = 1, ..., m$, come:

$$r_{f,X}(x) = \sum_{i=1}^{m} \alpha_i \phi(||x - x_i||), \tag{3.17}$$

dove $\alpha_1, ..., \alpha_m$ sono dei coefficienti in $\mathbb{R}$, $||x||$ è la norma di x, e ϕ è una funzione definita positiva; alcuni esempi della funzione ϕ sono :

- $\phi(r) = r^\beta$, con $\beta > 0$

- $\phi(r) = r^k \log(r)$, con $k \in \mathbb{N}$
- $\phi(r) = e^{-\alpha r^2}$, con $\alpha > 0$.

Esiste inoltre un teorema che afferma che se ϕ è una funzione definita positiva, allora esiste una e una sola funzione $r_{f,X}$ definita come in (3.17) che soddisfa le condizioni $r_{f,X}(x_i) = f(x_i)$ per ogni $i = 1, ..., m$. Il vantaggio di queste funzioni di base radiale risiede nel fatto che queste sono funzioni ben conosciute e studiate nell'ambito della teoria dell'approssimazione. Invece un problema al quale si va incontro utilizzando tali funzioni in un sistema di raccomandazione è che molto spesso lo spazio $\vec{c} \times \vec{s}$ non è un sottospazio di $\mathbb{R}^N$, in quanto gli attributi di $\vec{c}$ e $\vec{s}$ possono anche non essere numeri reali. Perciò una sfida per la ricerca è quella di estendere i metodi di base radiale dal dominio dei numeri reali ad altri domini per poi applicarli ai sistemi di raccomandazione.

Un ulteriore metodo che può essere utilizzato al fine di migliorare la qualità delle raccomandazioni è quello di prendere in considerazione anche alcune informazioni contestuali. Tipicamente i sistemi di raccomandazione operano su degli spazi bidimensionali ($Utente \times Oggetto$), ma in alcuni casi può essere utile fornire delle raccomandazioni basandosi anche su altre informazioni: ad esempio l'utilità di un prodotto può variare in base al periodo dell'anno. Risulta quindi utile considerare anche alcune informazioni contestuali come data, luogo etc, estendendo così le tradizionali raccomandazioni bidimensionali ($Utente \times Oggetto$) ad uno spazio multidimensionale.

A questo scopo è necessario definire la funzione di utilità in uno spazio multidimensionale $D_1 \times ... \times D_n$ [3]:

$$u : D_1 \times ... \times D_n \to R \tag{3.18}$$

Quindi il problema della raccomandazione viene ricondotto ad individuare per ogni tupla $(d_{j1}, ..., d_{jl}) \in D_{j1} \times ... \times D_{jl}$ (che rappresenta lo spazio relativo agli oggetti) la tupla $(d_{i1}, ..., d_{ik}) \in D_{i1} \times ... \times D_{ik}$ (lo spazio degli utenti) che massimizza l'utilità $u(d_1, ..., d_n)$, con $\{D_{i1}, ..., D_{ik}\} \cap \{D_{j1}, ..., D_{jl}\} = \emptyset$ e $l < n, k < n$, cioè :

$$\begin{aligned} &\forall (d_{j1}, ..., d_{jl}) \in D_{j1} \times ... \times D_{jl}, \\ &(d_{i1}, ..., d_{ik}) = \underset{\substack{(d'_{i1}, ..., d'_{ik}) \in D_{i1} \times ... \times D_{ik} \\ (d'_{j1}, ..., d'_{jl}) = (d_{j1}, ..., d_{jl})}}{\operatorname{argmax}} u(d'_1, ..., d'_n) \end{aligned} \tag{3.19}$$

Ad esempio, considerando un sistema di raccomandazione di film, bisogna considerare non solo le caratteristiche del film (d_1) e dell'utente

(d_2), ma anche informazioni contestuali come: (d_3) dove verrà visto il film (al cinema, a casa etc.), (d_4) con chi verrà visto (da solo, con amici etc.) e (d_5) quando verrà visto (ad esempio in che giorno della settimana). Ognuna delle componenti d_1, d_2, d_3, d_4, d_5 può essere definita da un vettore, e quindi la funzione di utilità $u(d_1, d_2, d_3, d_4, d_5)$ può diventare complessa da calcolare.

Prendendo in considerazione le informazioni d_i descritte sopra si può supporre che ognuna di queste sia un vettore le cui componenti sono numeri compresi tra 0 e 1 che rappresentano l'importanza che l'utente attribuisce a quella particolare caratteristica. Considerando ad esempio la caratteristica d_1 che determina il genere di film preferito dall'utente supponiamo che le sue componenti siano $d_{1,1} = 0.8$, $d_{1,2} = 0.7$ e $d_{1,3} = 0.5$ corrispondenti rispettivamente alle preferenze dell'utente nei confronti di un film d'azione ($d_{1,1}$), comico ($d_{1,2}$) e horror ($d_{1,3}$). Allo stesso modo si possono definire le altre caratteristiche d_i inserendo le informazioni descritte sopra e a questo punto, considerando il vettore formato dalle caratteristiche d_i, cioè $(d_1, d_2, d_3, d_4, d_5)$ e supponendo che la funzione di utilità sia definita come la misura di somiglianza del coseno, si cerca l'oggetto (anch'esso rappresentato da un vettore) che massimizza questa funzione.

Molti algoritmi usati per le raccomandazioni su spazi bidimensionali non possono essere estesi direttamente al caso multidimensionale, per questo [2] ha proposto un approccio basato sulla *riduzione della dimensione*: quando un utente richiede una raccomandazione fornendo dei criteri specifici, il sistema utilizza solamente i voti che sono pertinenti a quei criteri. Ricorrendo ancora all'esempio del sistema di raccomandazione di film, se un utente vuole vedere un film al cinema al sabato sera, i sistemi che adottano questo approccio utilizzeranno solo i voti disponibili relativi a film visti al cinema al sabato sera. In questo modo, dopo aver selezionato solamente gli oggetti che risultano essere rilevanti per la raccomandazione, questo approccio riconduce il problema della multidimensionalità allo spazio bidimensionale di *Utenti* e *Oggetti*.

Un altro approccio possibile per estendere le raccomandazioni al caso multidimensionale è quello di estendere i metodi Bayesiani: anzichè considerare solo i vettori z_i e w_j (rispettivamente le caratteristiche dell'utente e dell'oggetto) rappresentati da d_1 e d_2, si possono aggiungere altre dimensioni contestuali $d_3, ..., d_n$, con $d_i = (d_{i1}, ..., d_{ix_i})$ vettore contenente le caratteristiche per la dimensione D_i.

La maggior parte dei sistemi di raccomandazione trattano solamente voti basati su un singolo criterio, come ad esempio i voti relativi a film o libri. In altre applicazioni invece può essere utile basare i voti su più

criteri, si parla di *multicriteria in recommender systems* [3]; ad esempio in un sistema che raccomanda ristoranti si devono esaminare varie caratteristiche, tra le quali la qualità del cibo, del servizio e dell'arredamento. Alcune soluzioni per ottimizzare questo problema sono le seguenti:

- considerare una combinazione lineare dei vari criteri riducendo il problema ad un sistema a singolo criterio,
- considerare solo il criterio più importante e utilizzare gli altri criteri come vincoli,
- considerare consecutivamente un criterio alla volta convertendo la soluzione ottimale trovata ad un vincolo, e ripetere lo stesso procedimento per gli altri criteri.

Se si considera il secondo approccio applicato ad un sistema che deve raccomandare un ristorante r ad un utente c basandosi sui criteri forniti dall'utente di qualità del cibo $f_c(r)$, arredamento $d_c(r)$ e servizio $s_c(r)$, si può considerare $f_c(r)$ come criterio più importante, e usare gli altri come vincoli, cioè deve essere $d_c(r) > \alpha_c$ e $s_c(r) > \beta_c$, dove α_c e β_c sono i voti minimi forniti dall'utente relativi ad arredamento e servizio. Il problema è che normalmente non si conoscono i voti $d_c(r)$ e $s_c(r)$ per tutti i ristoranti, e quindi il sistema deve stimare i voti sconosciuti $d'_c(r)$ e $s'_c(r)$, per poi trovare tutti i ristoranti r che soddisfano le condizioni $d'_c(r) > \alpha_c$ e $s'_c(r) > \beta_c$. A questo punto si può cercare tra tutti questi ristoranti quello che ha il massimo punteggio per $f_c(r)$. Allo stesso modo però potrebbero non essere noti i punteggi $f_c(r)$ di tutti i ristoranti, e quindi bisogna ancora utilizzare un procedimento per stimare tali punteggi prima di fornire la raccomandazione.

Molti sistemi di raccomandazione sono invadenti nel senso che richiedono un feedback esplicito dell'utente e spesso ad un livello significativo di conivolgimento. Ad esempio prima di poter raccomandare degli articoli su un gruppo di discussione, è necessario acquisire i punteggi che l'utente ha attribuito agli articoli letti in precedenza. Alcuni sistemi di raccomandazione sono in grado di determinare tali punteggi in modo non invadente. Ad esempio il tempo che un utente impiega a leggere un articolo può essere usato al posto del voto che l'utente dovrebbe altrimenti fornire. Questi approcci non invadenti però risultano essere poco accurati e quindi non possono rimpiazzare completamente i voti forniti esplicitamente dall'utente. Perciò il problema principale che si deve affrontare in quest' ottica è quello di minimizzare l'invadenza mantenendo un buon livello di accuratezza. Un modo per ottenere questo risultato consiste nel determinare il numero ottimale di voti che un nuovo utente dovrebbe

fornire: prima di poter formulare delle raccomandazioni accurate, l'utente deve votare n oggetti e ognuno di questi voti richiede un costo C da parte dell'utente. Il costo totale sarà quindi $C \cdot n$, mentre ogni voto aggiuntivo fornito dall'utente comporterà un aumento dell'accuratezza della raccomandazione, portando un beneficio $B(n)$ all'utente. Si tratta quindi di determinare il valore ottimale di n che massimizza l'espressione $B(n) - C \cdot n$.

3.5 Efficienza

Tipicamente le prestazioni di un sistema di raccomandazione vengono misurate con delle metriche di *copertura* e *accuratezza* [3]. La prima misura la percentuale di oggetti per i quali il sistema è in grado di fornire delle predizioni, mentre la seconda misura la precisione della raccomandazione (cioè quanto l'oggetto raccomandato è stato effettivamente gradito dall'utente). L'accuratezza può essere calcolata con metodi statistici, principalmente confrontando i punteggi stimati dal sistema con quelli realmente forniti dagli utenti, calcolando ad esempio l'errore quadratico medio.

Questa metrica però può essere applicata solamente sugli oggetti che l'utente ha deciso di votare, e poichè un utente voterà principalmente gli oggetti che a lui piacciono, non viene testata l'abilità del sistema di raccomandare appropriatamente un oggetto che non appartiene alle preferenze dell'utente.

Inoltre, quando si misura l'efficienza della raccomandazione, si deve tener conto anche dell' *utilità* e della *qualità* della raccomandazione [24]. Ad esempio se un'applicazione che raccomanda articoli di un centro commerciale suggerisce agli utenti articoli che sicuramente loro compreranno (come ad esempio il pane) questo comporta un maggiore tasso di accuratezza del sistema poichè risulta che gli articoli raccomandati sono stati effettivamente comprati, anche se le raccomandazioni fornite non sono state utili per gli utenti.

Capitolo 4

Integrare Sistemi di Reputazione e Raccomandazione

Recentemente è stato effettuato uno studio [13] su come integrare i sistemi di raccomandazione e reputazione in modo da ottenere delle migliori raccomandazioni nei servizi online. Pur essendo basati su dei principi differenti, i sistemi di reputazione e raccomandazione hanno entrambi lo stesso scopo, ossia quello di aiutare l'utente fornendo delle informazioni per agevelore le sue decisioni. Infatti i sistemi di raccomandazione principalmente suggeriscono all'utente quali sono le risorse che lui non conosce ma alle quali molto probabilmente è interessato, basandosi sulle informazioni fornite da altri utenti di una comunità, come nel caso del collaborative filtering. Invece i sistemi di reputazione forniscono alla comunità delle informazioni riguardanti le risorse che l'utente conosce già. Si nota quindi che questi due sistemi sono simili tra loro e possono perciò essere combinati assieme ottenendo così un sistema in grado di fornire raccomandazioni più accurate.

La fusione dei due sistemi risulta però complicata poichè questi sono tra loro differenti, ad esempio i due sistemi possono utilizzare delle forme diverse di feedback e basare le proprie predizioni su criteri diversi. Inoltre non necessariamente i risultati che si ottengono dai due sistemi sono della stessa forma, ad esempio un sistema di reputazione può generare dei punteggi nella forma 1-5 stelle mentre un sistema di raccomandazione considera delle tuple del tipo $(d_1, ..., d_k)$ dove ogni componente è relativa ad una diversa caratteristica della risorsa. Prima di fondere i risultati dei due sistemi è quindi necessario renderli omogenei.

In [13] viene proposto un metodo con il quale uniformare i risultati

e poi combinarli: in primo luogo i risultati dei sistemi di reputazione e raccomandazione vengono trasformati in *opinioni soggettive* (si veda [13] per ulteriori dettagli) e poi si utilizza l'operatore CasMin (*Cascading Minimum Common Belief Fusion*, definito in [13]) per fondere i risultati ottenuti. Con questo operatore si può ottenere un valore alto di CasMin solo se entrambi i punteggi prodotti dai sistemi di raccomandazione e reputazione sono alti. In questo modo una risorsa verrà suggerita all'utente solo se è stata raccomandata dal sistema di raccomandazione con alta confidenza e ha un alto punteggio di reputazione.

Ne risulta quindi che la raccomandazione così ottenuta sarà molto più accurata rispetto a quella fornita da un sistema di raccomandazione isolato e si può quindi raggiungere una migliore qualità dei consigli che vengono distribuiti agli utenti, con una conseguente ottimizzazione dei servizi online.

Per comprendere meglio il funzionamento di tale metodo di seguito viene riportato l'esempio presentato in [13].

Considerato un sistema che deve fornire consigli riguardanti hotel, il sistema di raccomandazione deve tener traccia delle preferenze degli utenti, mentre quello di reputazione deve permettere agli utenti di votare gli hotel. Il sistema di reputazione produce dei punteggi di reputazione che possono essere trasformati in opinioni multinomiali, le quali a loro volta possono essere mappate in opinioni binomiali. Allo stesso modo il sistema di raccomandazione può rappresentare i voti utilizzando più dimensioni (relative ai vari criteri considerati nella formulazione del proprio giudizio) e quindi vengono prodotte delle opinioni multinomiali trasformate in opinioni binomiali. A questo punto l'operatore *CasMin* prende come input i valori così ottenuti, producendo un risultato che tiene conto dei consigli ottenuti da entrambi i sistemi di reputazione e raccomandazione.

Un esempio numerico è rappresentato nella Tabella 4.1 che illustra il risultato ottenuto fondendo le opinioni calcolate dai due sistemi utilizzando l'operatore *CasMin*: nella seconda colonna (a partire da sinistra) vengono elencati i punteggi ottenuti dal sistema di reputazione ($r(x_i)$ è il voto assegnato alla caratteristica x_i), mentre nella terza colonna viene rappresentata l'opinione multinomiale ottenuta dal punteggio di reputazione ($b(x_i)$ è il valore di belief che è stato calcolato a partire da $r(x_i)$, invece u_X è il valore di incertezza). Nella quarta colonna si possono vedere le opinioni binomiali ottenute dal punteggio di reputazione (b, d, u sono i valori di belief, disbelief e uncertainty per l'opinione binomiale). Nella quinta colonna vengono raggruppate le opinioni binomiali ottenute dal sistema di raccomandazione e infine nella sesta colonna viene presen-

tato il risultato ottenuto dall'operatore *CasMin*.

Nel caso degli Hotel I e II i punteggi (binomiali) del sistema di reputazione e quello di raccomandazione sono contrastanti: l'Hotel I ha un alto punteggio di reputazione ($b = 0.81$) ma non si abbina alle preferenze dell'utente ($b = 0.1$), mentre nel secondo caso succede l'opposto. Quindi quando si combinano i risultati tramite l'operatore *CasMin* si ottiene un basso valore di belief (rispettivamente $b = 0.30$ e $b = 0.19$) e un alto valore di disbelief ($d = 0.70$ e $d = 0.61$). Questo significa che gli Hotel I e II non sono adatti per essere raccomandati all'utente, in quanto il primo non riflette le preferenze dell'utente, mentre il secondo pur avvicinandosi alle sue preferenze ha una cattiva reputazione

Invece nel caso dell'Hotel III poichè entrambi i valori di reputazione e raccomandazione sono elevati (rispettivamente $b = 0.81$ e $b = 0.7$), applicando l'operatore *CasMin* si ottiene un alto valore di belief ($b = 0.81$) e un basso valore di disbelief ($d = 0.19$). A questo punto si può decidere di raccomandare l'Hotel III, poichè è quello che più si avvicina alle preferenze dell'utente e gode di una buona reputazione.

Hotel	Punteggi di Reputazione	Punt. di Rep. Multinomiali	Punt. di Rep. Binomiali	Valore Racc.	*CasMin*
Hotel I	$r(x_5) = 50$ $r(x_4) = 10$ $r(x_3) = 10$ $r(x_2) = 0$ $r(x_1) = 5$	$b(x_5) = 0.65$ $b(x_4) = 0.13$ $b(x_3) = 0.13$ $b(x_2) = 0.00$ $b(x_1) = 0.06$ $u_X = 0.03$	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.1$ $d = 0.7$ $u = 0.2$	$b = 0.30$ $d = 0.70$ $u = 0.00$
Hotel II	$r(x_5) = 5$ $r(x_4) = 0$ $r(x_3) = 10$ $r(x_2) = 10$ $r(x_1) = 50$	$b(x_5) = 0.06$ $b(x_4) = 0.00$ $b(x_3) = 0.13$ $b(x_2) = 0.13$ $b(x_1) = 0.65$ $u_X = 0.03$	$b = 0.16$ $d = 0.81$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.19$ $d = 0.61$ $u = 0.20$
Hotel III	$r(x_5) = 50$ $r(x_4) = 10$ $r(x_3) = 10$ $r(x_2) = 0$ $r(x_1) = 5$	$b(x_5) = 0.65$ $b(x_4) = 0.13$ $b(x_3) = 0.13$ $b(x_2) = 0.00$ $b(x_1) = 0.06$ $u_X = 0.03$	$b = 0.81$ $d = 0.16$ $u = 0.03$	$b = 0.7$ $d = 0.1$ $u = 0.2$	$b = 0.81$ $d = 0.19$ $u = 0.00$

Tabella 4.1: esempio numerico della fusione con *CasMin*.

Capitolo 5

Conclusioni

In questa tesi sono stati esaminati i sistemi di reputazione e raccomandazione, i quali vanno inseriti nell'ambito dei sistemi di supporto alle decisioni. Sono stati presentati i pro e i contro di tali sistemi cercando di fornire delle soluzioni per risolvere alcune problematiche che li affliggono. Inoltre è stato presentato un modo per integrare i duei tipi di sistemi allo scopo di ottenere dei migliori consigli nelle applicazioni online.

I sistemi di reputazione possono essere utilizzati in molti tipi di applicazione e calcolano i punteggi di reputazione utilizzando diversi metodi; sarà quindi necessario determinare il sistema che si adatta meglio al contesto e all'applicazione all'interno della quale deve essere utilizzato. La crescente letteratura riguardante i sistemi di reputazione per le transazioni in Internet e lo sviluppo di questo tipo di sistemi in molteplici applicazioni commerciali sono un indice dell'importanza di queste tecnologie.

Nell'ultimo decennio i sistemi di raccomandazione sono andati incontro a numerosi progressi, in seguito allo sviluppo di vari tipi di sistemi: basati sul contenuto, sistemi collaborativi e sistemi ibridi. Tuttavia i sistemi della generazione odierna presentano ancora molte imperfezioni che possono essere risolte. Infatti i sistemi attuali possono essere migliorati in vari modi, ad esempio utilizzando dei profili che incorporano un maggior numero di informazioni riguardanti gli utenti e gli oggetti, integrando altre informazioni contestuali al processo di raccomandazione, e implementando dei sistemi meno invadenti e più flessibili in grado di formulare le raccomandazioni in base ai bisogni dell'utente.

Inoltre un buon metodo per migliorare la qualità delle raccomandazioni fornite dalle applicazioni online consiste nell'integrare i due tipi di sistemi: così facendo si possono fornire delle raccomandazioni molto più accurate e affidabili rispetto a quelle che si ottengono solamente con

un sistema di raccomandazione. Questo porterebbe ad una migliore efficienza, con una conseguente diffusione più ampia di tali sistemi nelle applicazioni online, sia commerciali che sociali.

Capitolo 6

Sviluppi Futuri

Un interessante sviluppo futuro dei sistemi di reputazione e raccomandazione potrebbe essere quello di integrare questi sistemi all'interno dei Social Network in modo da affinare e personalizzare ulteriormente il processo di raccomandazione. In quest'ottica sono già stati intrapresi degli studi [4, 10] e sembra essere una direzione molto promettente che potrebbe permettere una grande diffusione di questi sistemi. Effettuando questa integrazione si potrebbero includere nel processo di raccomandazione anche informazioni contestuali prelevate in maniera automatica, magari monitorando l'attività e le preferenze dell'utente e dei suoi amici nei Social Network. Così facendo i dati potrebbero essere raccolti in modo del tutto trasparente all'utente, il quale non dovrà essere esplicitamente interrogato, aumentando anche l'usabilità dell'intero sistema. Altri sviluppi che potrebbero migliorare notevolmente l'usabilità di tali sistemi consistono nel definire delle nuove metriche con le quali poter calcolare la qualità e l'efficienza delle raccomandazioni e l'implementazione di nuovi metodi di integrazione dei sistemi di reputazione e raccomandazione sempre più accurati ed efficienti.

Si auspica che le proposte di sviluppi futuri possano fornire degli spunti per migliorare le tecnologie delle future generazioni, in modo da avere una migliore integrazione nelle applicazioni commerciali e sociali dei sistemi di reputazione e raccomandazione.

Bibliografia

[1] A. Abdul-Rahman and S. Hailes. "Supporting Trust in Virtual Communities". In Proceedings of the Hawaii International Conference on System Sciences, Maui, Hawaii, 4-7 January 2000.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach", ACM Trans. Information Systems, vol. 23, no. 1, Jan. 2005.

[3] G. Adomavicius and A. Tuzhilin. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". IEEE Transactions on Knowledge and Data Engineering, Vol. 17, no. 6, June 2005.

[4] K. Al Falahi, N. Mavridis, and Y. Atif. "Social Networks and Recommender Systems: A World of Current and Future Synergies". Computational Social Networks: Tools, Chapter 18, Perspectives and Applications, Springer-Verlag London 2012.

[5] A. Ansari, S. Essegaier, and R. Kohli. "Internet Recommendations Systems". J. Marketing Research, pp. 363-375, Aug. 2000.

[6] M. Balabanovic and Y. Shoham. "Fab: Content-Based, Collaborative Recommendation". Comm. ACM, vol. 40, no. 3, pp. 66-72, 1997.

[7] J.S. Breese, D. Heckerman, and C. Kadie. "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". Proc. 14th Conf. Uncertainty in Artificial Intelligence, July 1998.

[8] M.D. Buhmann. "Approximation and Interpolation with Radial Functions". Multivariate Approximation and Applications, N. Dyn, D. Leviatan, D. Levin, and A. Pinkus, eds., Cambridge Univ. Press, 2001.

[9] D. Gambetta. “Can We Trust Trust?”. In D. Gambetta, editor, Trust: Making and Breaking Cooperative Relations, pages 213238. Basil Blackwell. Oxford, 1990.

[10] J. He, W. W. Chu. “A Social Network-Based Recommender System (SNRS)”. Data Mining for Social Network Data Annals of Information Systems, Volume 12, pp 47-74, Springer US, 2010.

[11] A. Jøsang. “A Logic for Uncertain Probabilities”. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, June 2001.

[12] A. Jøsang. “Trust-Based Decision Making for Electronic Transactions”. In L. Yngstrom and T. Svensson, editors, Proceedings of the 4th Nordic Workshop on Secure Computer Systems (NORDSEC99). Stockholm University, Sweden, 1999.

[13] A. Jøsang, G. Guo, M. S. Pini, F. Santini, Y. Xu. “Combining Recommender and Reputation Systems to Produce Better Online Advice”. Proceedings of the 10th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2013), Barcelona, Spain, Springer LNAI, full paper, 2013.

[14] A. Jøsang and R. Ismail. “The Beta Reputation System”. In Proceedings of the 15th Bled Electronic Commerce Conference, June 2002.

[15] A. Jøsang, R. Ismail, and C. Boyd. “A Survey of Trust and Reputation Systems for Online Service Provision”. Decision Support Systems, 43(2), pages 618-644, March 2007.

[16] R.J. Mooney, P.N. Bennett, and L. Roy. “Book Recommending Using Text Categorization with Extracted Information”. Proc. Recommender Systems Papers from 1998 Workshop, Technical Report WS-98-08, 1998.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. “The PageRank Citation Ranking: Bringing Order to the Web”. Technical report, Stanford Digital Library Technologies Project, 1998.

[18] M. Pazzani. “A Framework for Collaborative, Content-Based, and Demographic Filtering”. Artificial Intelligence Rev., pp. 393-408, December 1999.

[19] M. Pazzani and D. Billsus. "Learning and Revising User Profiles: The Identification of Interesting Web Sites". Machine Learning, vol. 27, pp. 313-331, 1997.

[20] P. Resnick and R. Zeckhauser. "Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System". In M.R. Baye, editor, The Economics of the Internet and E-Commerce, volume 11 of Advances in Applied Microeconomics. Elsevier Science, 2002.

[21] P. Resnick, R. Zeckhauser, R. Friedman, and K. Kuwabara. "Reputation Systems". Communications of the ACM, December 2000.

[22] U. Shardanand and P. Maes. "Social Information Filtering: Algorithms for Automating Word of Mouth ". Proc. Conf. Human Factors in Computing Systems, 1995.

[23] I. Soboroff and C. Nicholas. "Combining Content and Collaboration in Text Filtering". Proc. Intl Joint Conf. Artificial Intelligence Workshop: Machine Learning for Information Filtering, August 1999.

[24] Y. Yang and B. Padmanabhan. "On Evaluating Online Personalization". Proc. Workshop Information Technology and Systems, pp. 35-41, December 2001.

[25] B. Yu & M.P. Singh. "An Evidential Model of Distributed Reputation Management". In Proceedings of the First Int. Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS). ACM, July 2002.