

**ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE
CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE**

I SISTEMI DI RACCOMANDAZIONE IN AMBITO E-LEARNING

**Relazione finale in:
Programmazione**

**Relatore:
Prof. ssa Antonella Carbonaro**

**Presentata da:
Simone Di Nardo Di Maio**

**Sessione II
Anno Accademico 2012/2013**

a Mamma e Papà,

a Boo..

Sommaro

Introduzione	1
Capitolo 1 – Sistemi di Raccomandazione.....	3
1.1 Metodi puri	4
1.1.1 Metodo Collaborative	4
Analisi del metodo	5
1.1.2 Metodo Demographic.....	6
Analisi del metodo	7
1.1.3 Metodo Content-Based	7
Analisi del metodo	8
1.1.4 Metodo Utility-Based	8
Analisi del metodo	9
1.1.5 Metodo Knowledge-Based	9
Analisi del metodo	10
1.1.6 Considerazioni sui metodi puri	10
1.2 Approcci ibridi.....	12
1.2.1 Approccio “Weighted”	12
1.2.2 Approccio “Mixed”	12
1.2.3 Approccio “Features Combination”	13
1.2.4 Approccio “Cascade”.....	13
1.2.5 Approccio “Feature Argomentation”	14
1.2.6 Approccio “Meta-Level”	15
1.2.7 Approcci ibridi a confronto.....	15
Capitolo 2 – Applicazioni pratiche.....	19
2.1 RSSEs – Recommendation Systems for Software Engineering	19
2.1.1 eRose	20

2.1.2 Strathcona	21
2.1.3 Suade	24
2.2 ADDRESS – Asan Adverse Drug Reaction EMR Surveillance System ..	25
2.3 Sistemi di raccomandazione Firewall	28
2.4 Sistemi di raccomandazione per l'e-commerce turistico.....	30
2.4.1 Web Data Mining	30
2.4.2 Sistemi di raccomandazione e-commerce	31
2.4.3 Raccomandazione tramite metodo Association Rule	32
2.4.4 Applicazione in ambito turistico dei sistemi di raccomandazione per e-commerce	32
2.4.5 Realizzazione del sistema di raccomandazione	34
Capitolo 3 – Tecniche di raccomandazione e-Learning.....	37
3.1 Il contesto tecnologico	37
3.1.1 Computer Based Learning.....	38
3.1.2 Formazione a distanza	39
3.1.3 Piattaforme di e-Learning	40
3.1.4 Learning Object	40
3.1.4.1 Standard per i Learning Object.....	41
3.1.4.2 Learning Object Repository	41
3.1.5 Analisi del contesto tecnologico.....	42
3.2 Il contesto pedagogico	43
3.2.1 Obiettivi didattici	44
3.2.2 Forma del contenuto.....	44
3.2.3 Strategie di insegnamento	45
3.2.4 Stili cognitivi – Modello di Felder.....	46
3.3 Sistema di raccomandazione e modello studente	48
3.3.1 Rappresentazione della conoscenza del dominio	49

3.4 Metodo di raccomandazione Cascade.....	52
3.4.1 Sistema di raccomandazione LMS	52
3.4.2 Processo di raccomandazione	54
3.4.3 Formulazione del suggerimento	56
3.3.4 Precisione nella formulazione del suggerimento	57
3.5 Tecnica Rule-based per la raccomandazione in ambito e-Learning.....	58
3.5.1 La tecnica Rule-based	59
3.6 Moodle	62
3.6.1 Approccio di base.....	62
3.6.2 Dynamic Clustering	63
3.6.3 Implementazione di Moodle	64
3.7 Sistemi di raccomandazione per la scelta di obiettivi didattici	65
3.7.1 Intelligent Web Teacher	65
3.7.2 Descrizione del processo	66
3.7.3 Valutazione dell'utilità del concetto.....	67
3.7.4 Definizione degli ULLGs.....	68
Capitolo 4 - Privacy	71
4.1 Pistis	72
4.1.1 Definizione del problema.....	73
4.1.2 Privacy Attack Model.....	74
4.1.3 Funzionamento di Pistis	76
4.1.4 Risultati di Pistis	77
Conclusione	79
Bibliografia	81
Ringraziamenti	87

Introduzione

Gli anni '90 hanno visto l'espandersi di Internet e un abbattimento dei costi della tecnologia tali da permettere a molte famiglie di possedere una postazione di accesso alla rete globale.

Contestualmente sono nate le prime piattaforme che permettevano un rapido accesso alla cultura, ma sempre limitate dalla tecnologia del tempo e da connessioni con banda molto ridotte. Risorse erano reperibili dopo ore di ricerche sul web, senza avere la possibilità di catalogare o di riunire tutte quelle disponibili.

Nell'ultimo decennio, invece, sono nati progetti come Wikipedia ®, l'enciclopedia libera, che permette un rapido accesso a nozioni e fatti a costi quasi nulli. Motori di ricerca come Google ® e Yahoo ® permettono di reperire milioni di fonti e di testi su un singolo argomento. È sbalorditivo come si sia passati letteralmente da un eccesso ad un altro.

L'abbondanza di informazioni spaventa i lettori: se qualche anno fa, ricercando del materiale, uno studente trovava dieci risorse le leggeva. Adesso, trovandone 10 milioni, si demoralizza e smette di cercare.

Da questo problema nascono i primi sistemi di raccomandazione, sistemi software integrati in portali web, strumenti di sviluppo, che aiutano l'utente a reperire esclusivamente le risorse che più sono pertinenti alle sue esigenze.

I sistemi di raccomandazione per come li conosciamo nascono alla fine del XX secolo, e si sono evoluti fino ai giorni nostri approcciandosi a numerosi campi, tra i quali analizzeremo l'ingegneria del software, la medicina, la gestione delle reti aziendali e infine, come argomento focale della tesi, l'e-Learning.

Dopo una rapida panoramica sullo stato dell'arte dei sistemi di raccomandazione al giorno d'oggi, scorrendo velocemente tra metodi puri e metodi ibridi ottenuti come combinazione dei primi, analizzeremo varie applicazioni pratiche per dare un'idea al lettore di quanto possano essere vari i settori di utilizzo di questi software.

Tratteremo nello specifico il funzionamento di varie tecniche per la raccomandazione in ambito e-Learning, analizzando tutte le problematiche che distinguono questo settore da tutti gli altri. Nello specifico, dedicheremo un'intera sezione alla descrizione della psicologia dello studente, e su come capire il suo profilo cognitivo aiuti a suggerire al meglio la giusta risorsa da apprendere nel modo più corretto.

È doveroso, infine, parlare di privacy: come vedremo nel primo capitolo, i sistemi di raccomandazione utilizzano al massimo dati sensibili degli utenti al fine di fornire un suggerimento il più accurato possibile. Ma come possiamo tutelarli contro intrusioni e quindi contro violazioni della privacy?

L'obiettivo di questa tesi è quindi quello di presentare al meglio lo stato attuale dei sistemi di raccomandazione in ambito e-Learning e non solo, in modo da costituire un riferimento chiaro, semplice ma completo per chiunque si volesse affacciare a questo straordinario ed affascinante mondo della raccomandazione on line.

Capitolo 1 – Sistemi di Raccomandazione

La prima definizione di “sistema di raccomandazione” ci viene fornita nel 1997 dagli studiosi Resnick e Varian [1], come sistemi capaci di aggregare e rielaborare input espliciti forniti dall’utente relativi ai loro interessi in modo da orientare la sua ricerca verso insiemi o portali che rispecchino le loro preferenze. Attualmente vengono visti in maniera più ampia: un’altra definizione di *recommending system* ci viene fornita da Robin Burke, secondo cui è un insieme di sistemi, tecniche, algoritmi che producono suggerimenti per l’utente come risultato della propria computazione, o che guidano l’utente in modo personalizzato in un ampio spazio di opzioni. [2]. Una definizione formale invece può essere data in questi termini [3]: Sia C l’insieme degli $m \in \mathbb{N}$ utenti del sistema, ed I l’insieme degli $n \in \mathbb{N}$ oggetti che possono essere raccomandati. Viene definita *funzione di utilità* $u: C \times I \rightarrow \mathbb{R}$ che misura quanto un oggetto i sia utile ad un dato utente c . \mathbb{R} è un insieme totalmente ordinato ed i suoi elementi rappresentano quantitativamente l’utilità di un oggetto per un utente.

Sostanzialmente, i sistemi di raccomandazione sono un’evoluzione dei motori di ricerca classici; questi ultimi forniscono una risposta al quesito dell’utente controllando le corrispondenze all’interno del proprio database, ordinandole in base al valore di matching. La novità introdotta dai sistemi di raccomandazione sta nel riuscire a fornire una rappresentazione personalizzata del sistema, sfruttando i feedback dell’utente per fornire suggerimenti più pertinenti.

Le tecniche che permettono di realizzare i sistemi di raccomandazione vengono suddivise in due categorie:

- *Metodi puri*;
- *Metodi ibridi*;

Nel seguito del capitolo verranno analizzati questi metodi, focalizzandosi principalmente sugli aspetti negativi di ciascun approccio, per poi completare l’excursus andando a studiare come i metodi ibridi, ottenuti come combinazione dei metodi puri, riescono a ridurre le negatività dei singoli metodi.

1.1 Metodi puri

I metodi di raccomandazione possono essere classificati sulla base di numerose direttrici; l'interazione con l'utente, la capacità di apprendimento, le tecniche di inferenza adottate per fornire i suggerimenti e molto altro ancora.

Partendo dai dati risidenti nel sistema, possiamo fornire una distinzione tra dati di background (l'insieme dei dati gestiti direttamente dal sistema, conosciuti prima dell'inizio del processo di raccomandazione) e dati di input (forniti direttamente dall'utente al fine di soddisfare la propria richiesta). L'algoritmo di raccomandazione combina questi due tipi di dati cercando di fornire il suggerimento.

Sulla base di questi tre elementi (dati di input, dati di background, algoritmo) possiamo distinguere i seguenti cinque metodi puri:

- *Collaborative*;
- *Demographic*;
- *Content-Based*;
- *Utility-Based*;
- *Knowledge-Based*;

Prima di addentrarci nell'analisi dei metodi, diamo la definizione di problema di *Ramp-up*: è una difficoltà che nasce al momento dell'inserimento di un nuovo oggetto o di un nuovo utente. Il suggerimento per un oggetto che ha poche recensioni all'interno del database della raccomandazione, sarà poco efficiente. Stesso discorso vale per un nuovo utente, del quale non si conoscono gusti e preferenze .[14]

1.1.1 Metodo Collaborative

Il metodo collaborativo è uno dei più utilizzati attualmente. L'idea di base è quella di aggregare le valutazioni o le raccomandazioni sugli oggetti da suggerire, sulla base delle somiglianze fra gli utenti e delle loro valutazioni, al fine di generare in accordo con esse nuovi suggerimenti. [4]

Il metodo collaborativo, generalmente, crea un profilo utente per mezzo di un vettore continuamente aggiornato durante l'interazione dell'utente col sistema. Le valutazioni possono essere booleane (piace/non piace) o a valori floating

point indicando il grado di preferenza. Il suggerimento viene fornito tramite un conteggio pesato sulla data di inserimento della preferenza – valutazioni recenti hanno pesi maggiori.

Prendendo ad esempio un negozio musicale, secondo il metodo collaborativo, ad un cliente patito di *Janis Joplin* potrebbe essere consigliato “*Are you experienced?*” di *Jimi Hendrix* perché altri fan di *Janis Joplin* hanno acquistato quell’album.

Tra i sistemi collaborativi si sono ulteriormente distinte diverse categorie a seconda dell’algoritmo utilizzato per esplorare le connessioni tra gli utenti. Distinguiamo:

- algoritmi *Memory-Based*: in questa categoria rientrano gli algoritmi che si basano sulla storia delle valutazioni degli utenti del sistema per predire valutazioni future
- algoritmi *Model-Based*: in questa categoria rientrano gli algoritmi che sfruttano la storia del sistema per apprendere un modello che poi è utilizzato per generare raccomandazioni. In questa categoria rientrano sistemi che fanno uso di reti di Bayes, reti neurali o tecniche di clustering.

Analisi del metodo

Sfruttando le valutazioni dei singoli utenti, si avranno sicuramente dei problemi di dispersione dei dati, oppure se pochi utenti hanno fornito un feedback sugli stessi oggetti.

Definiamo *Cold-Start* il problema che viene riscontrato in un sistema collaborativo nei primi tempi di vita, in quanto si ha disponibile un numero di valutazioni inferiore rispetto a quelli di un sistema a regime. Un aspetto meno centrale ma pur sempre importante è quello che affligge gli oggetti più comuni del dominio: quelli che vengono scelti con più frequenza, un po’ come le banane al mercato che almeno una volta nella vita vengono acquistate da tutti. Ciò porta a raccomandare sempre e per tutti gli utenti tali oggetti, anche se non rispecchiano palesemente gli interessi di chi si avvale del sistema. Data l’affinità della situazione con quella delle banane al mercato, questo problema è stato ribattezzato *Banana Problem*.

Il problema della dispersione rischia di diventare un vero impedimento in domini come il news filtering¹ in cui il tasso di espansione dello spazio degli oggetti può essere notevolmente più alto rispetto alla crescita dello spazio delle valutazioni.

D'altro canto, è un metodo straordinariamente adattivo, più viene utilizzato e maggiore sarà la precisione del suggerimento. Inoltre l'implementazione dell'algoritmo è totalmente avulsa dal dominio in cui è applicato: questo rende la tecnica sufficientemente astratta da poter essere applicata in contesti diversi.

1.1.2 Metodo Demographic

Il metodo demografico classifica l'utente sulla base dei suoi attributi personali (altezza, provenienza) ed elabora il suggerimento in funzione della classe demografica a cui viene assegnato. Il primo esempio di sistema demografico chiamato *Grundy* venne presentato nel 1974 da Rich [5]. Esso suggeriva libri sulla base delle informazioni personali raccolte per mezzo di un dialogo interattivo con l'utente.

Questo metodo è stato utilizzato in seguito in altri sistemi [6] anche se con scarsa diffusione in quanto presenta punti di debolezza di difficile superamento. Il metodo demografico non fornisce le modalità con le quali verranno raccolti i dati. In alcuni casi [7] [8] sono stati utilizzati questionari per la classificazione utente, in altri venivano inferiti mediante tecniche di machine learning² [9][10]. Quanto detto vale anche per la rappresentazione delle informazioni che costituiranno il modello utente. Ad esempio, nei sistemi semplici vengono utilizzate coppie attributo-valore definite a priori, in sistemi più evoluti l'estrazione delle caratteristiche per la classificazione dell'utente viene gestita da un algoritmo di machine learning.

¹ Sistemi software che permettono di visualizzare principalmente notizie affini alla ricerca dell'utente

² Ramo dello studio dell'Intelligenza Artificiale, consiste nella costruzione di un sistema capace di apprendere da un insieme di dati.

Analisi del metodo

I sistemi di raccomandazione demografici non sono soggetti al problema del *new user* in quanto forniscono suggerimenti valutando le informazioni demografiche inserite.

Il limite principale consiste nella difficoltà di reperire le informazioni rilevanti al fine del rating, sia per la crescente sensibilizzazione sulla privacy (argomento che verrà trattato nel quarto capitolo) e il conseguente scetticismo degli utenti nel fornire informazioni personali in un form on-line. Un ulteriore punto di debolezza consiste nell'impossibilità di fornire suggerimenti *cross-genre*³.

I sistemi demografici puri sono piuttosto rari, in quanto i dati sensibili di cui ha bisogno per tracciare il profilo utente sono proprio quelli che gli utenti tendono a non fornire.

1.1.3 Metodo Content-Based

Il metodo content-Based nasce come evoluzione delle tecniche di information filtering [13]. In un sistema content-Based le feature che definiscono gli oggetti di interesse sono gli elementi dell'oggetto stesso. Ad esempio, ritornando all'esempio del negozio di musicale, ad un cliente che desidera acquistare un album dei *Rolling Stones* potrebbe essere suggerito uno dei *Beatles* dal momento che sono entrambe band degli anni '80.

In generale, ad ogni oggetto $i \in I$ viene assegnato un *Profilo(i)* ossia un insieme di tutte le caratteristiche capaci di riassumerne il contenuto. All'atto della raccomandazione viene generato un profilo utente basato sugli attributi degli oggetti che ha preferito in passato, denominato *ProfiloContentBased(c)*. L'utilità u per un oggetto i per il cliente c viene calcolata in base alla "somiglianza" tra i due diversi profili. In letteratura esistono esempi di creazione del *ProfiloContentBased(c)* tramite alberi di decisione [11], reti neurali [8] o rappresentazioni vettoriali[14].

³ Possibilità di suggerire oggetti anche di ambito diverso tra loro (documenti storici del XVIII secolo e XIX secolo)

Analisi del metodo

In questo tipo di approccio le predizioni tendono ad essere sovraspecializzate e, probabilmente, poco interessanti. Basandosi sulla storia del cliente, i suggerimenti ricalcherebbero troppo fedelmente le preferenze passate e non hanno quell'elemento di imprevedibilità che ci si aspetta da un sistema di raccomandazione. Questo può portare a ritenere il sistema poco utile, vista la prevedibilità dei suggerimenti. Per rendersi conto di quanto possa pesare questo problema, prendiamo ad esempio il negozio musicale citato precedentemente: ad un appassionato di *Bob Marley* verranno consigliati esclusivamente album dello stesso cantante; per arrivare allo stesso risultato sarebbe stato sufficiente consultare una discografia, il che rende il sistema di raccomandazione pressoché inutile.

Altra debolezza del metodo consiste nella necessità di descrivere ogni oggetto esclusivamente per le caratteristiche che lo compongono, il che rende il metodo strettamente dipendente dalla quantità di dati descrittivi disponibili.

Attualmente il proliferare dei linguaggi di meta-datazione per la descrizione delle risorse multimediali [14][15] ha permesso a questo metodo di poter essere applicato in maniera più proficua. Ciononostante, persino in presenza di dati descrittivi, alcuni esperimenti hanno certificato migliori performance di un sistema collaborativo rispetto ad un content-Based.

1.1.4 Metodo Utility-Based

Il metodo Utility-Based non tenta di costruire generalizzazioni a lungo termine dei propri utenti, ma suggerisce gli oggetti più opportuni confrontando, per mezzo di una funzione di utilità, le esigenze dell'utente con l'insieme delle possibili opzioni in grado di soddisfarle.[2]

Il problema principale è, quindi, quello di creare per ogni utente la funzione capace di stabilire quale oggetto massimizza la soddisfazione dell'utente. A tal fine, per calcolare la migliore corrispondenza, sono impiegate tecniche del tipo *constraint satisfaction* nel caso di feature simboliche o tecniche basate su funzioni matematiche nel caso di feature definite in \mathfrak{R} [15].

Il vantaggio del metodo Utility-Based sta nel fatto che esso è in grado di gestire attributi esterni al prodotto, come per esempio l'affidabilità del venditore o la disponibilità (ad esempio nell'e-commerce) del prodotto nel calcolo della funzione di utilità. Ciò permette di inserire ulteriori gradi di libertà che con i metodi precedentemente esaminati non era possibile considerare.

Analisi del metodo

Il problema principale di questo metodo risiede nel fatto che la funzione di utilità, scegliendo quale peso dare a quale caratteristica, viene scelta dall'utente che potrebbe sottovalutare questo compito, dando priorità, ad esempio, alla disponibilità di un prodotto a discapito della sua effettiva qualità. Un ulteriore punto di debolezza è dato dalla possibilità di un cambiamento delle preferenze dell'utente, che richiederebbe una rivoluzione della funzione di utilità.

Il metodo Utility-Based è totalmente insensibile al ramp-up, e permette di inserire come parametro della funzione di utilità feature differenti rispetto alle caratteristiche intrinseche dell'oggetto da raccomandare. Questa flessibilità viene pagata in termini di interazione con l'utente: potrebbe dover compilare un questionario per indicare le proprie preferenze oppure costruire direttamente la funzione di utilità [16].

1.1.5 Metodo Knowledge-Based

I sistemi di raccomandazione Knowledge-Based lavorano cercando di suggerire oggetti partendo dai dati a disposizione, dalle esigenze e dalle preferenze di ciascun utente. Questa tipologia di approccio si differenzia dalle altre in quanto possiede la conoscenza funzionale. Possiedono regole, espresse in forma esplicita, che permettono di stabilire come un particolare oggetto incontra le esigenze dell'utente e pertanto possono ragionare di conseguenza per fornire il suggerimento più opportuno.

Il profilo utente può essere qualunque rappresentazione di conoscenza che supporta questo tipo di inferenza. Nel caso più semplice potrebbe essere la richiesta dell'utente, in casi più complessi una rappresentazione maggiormente

dettagliata delle sue esigenze. In letteratura sono spesso utilizzate tecniche di ragionamento basate sulla prima logica, anche se la tecnica utilizzata è strettamente legata alla rappresentazione della conoscenza utilizzata [17][18].

Analisi del metodo

Il metodo Knowledge-Based, come tutti gli approcci basati sulla rappresentazione esplicita della conoscenza, presenta la difficoltà della fase di knowledge-acquisition.

La conoscenza necessaria riguarda tre ambiti differenti:

- *catalog knowledge*, relativa agli oggetti suggeriti e alle loro features;
- *functional knowledge*, la conoscenza che consente al sistema di mappare gli oggetti da suggerire con le caratteristiche dell'utente;
- *user knowledge*, la conoscenza sull'utente, acquisita su base demografica o specifica [2];

Eccezion fatta per questa fase molto dispendiosa, i metodi Knowledge-Based presentano numerosi vantaggi che non sono presenti negli altri approcci fin qui descritti.

Sono molto adatti ad un utente casuale, in quanto non si basano sul confronto delle valutazioni dell'utente: detto ciò è palese che sono insensibili al problema del ramp-up. In secondo luogo non necessitano di un periodo di start-up in cui viene offerto un servizio di scarsa qualità ed infine possono fornire suggerimento su domini di conoscenza anche molto ampi, operando con tecniche di inferenza di diverso tipo: induzione, deduzione, analogia ecc.

1.1.6 Considerazioni sui metodi puri

Come si può facilmente notare, possiamo distinguere questi cinque approcci tra quelli con apprendimento automatico (Collaborative, Demographic, Content-Based) e quelli con apprendimento tramite interazione con l'utente (Utility-Based, Knowledge-Based) .

I primi soffrono il ramp-up problem in una delle sue forme, con l'effetto collaterale di escludere utenti casuali dal ricevere i vantaggi dei suggerimenti. Basandosi su dati storicizzati, un ulteriore problema risiede nella lentezza di

aggiornamento dei profili: un utente che ha sempre letto notizie sportive farà fatica a ricevere notizie di politica qualora cambiasse i propri gusti, fino a quando il sistema non memorizzerà abbastanza feedback negativi sulle notizie sportive.

Alcune varianti dei metodi ad apprendimento automatico permettono di assegnare un peso minore a valutazioni vecchie, ma ciò comporta il rischio di perdere informazioni importanti sulle abitudini dell'utente se esercitate sporadicamente.

I metodi Knowledge-Based e Utility-Based, al contrario, rispondono alle esigenze immediate dell'utente e non necessitano di alcuna fase di start-up in cui le risposte del sistema di raccomandazione potrebbero distaccarsi molto dall'obiettivo reale. D'altro canto, il metodo Utility-Based può presentare difficoltà per gli utenti casuali che non intendono perdere tempo personalizzando il proprio profilo o rispondendo a un lungo questionario.

1.2 Approcci ibridi

I metodi di raccomandazione descritti precedentemente presentano dei punti di debolezza tipici per cui è necessario combinarli al fine di minimizzare gli svantaggi derivanti da ciascun metodo.

La combinazione può avvenire in diversi modi, e il metodo risultante è detto ibrido.

1.2.1 Approccio “Weighted”

La modalità Weighted (pesata), prevede che l’oggetto venga suggerito in base alla valutazione di tutti i sistemi di raccomandazione presenti. Supponendo che ci siano n metodi sul sistema, questa modalità mette insieme i suggerimenti provenienti da ciascun metodo per mezzo di una combinazione lineare dei punteggi ottenuti da ciascuna tecnica.

Il vantaggio consiste essenzialmente nella semplicità con cui il processo è messo in atto e nella possibilità di variare il peso di ciascun metodo, modificando il coefficiente con il quale partecipa alla combinazione lineare.

L’assunzione implicita in questa modalità è che ciascun metodo sia più o meno uniforme nello spazio degli oggetti. Ad esempio, un metodo Collaborative, in un dominio con poche valutazioni, dovrà avere un peso inferiore agli altri metodi utilizzati.

1.2.2 Approccio “Mixed”

Questo tipo di approccio prevede che i suggerimenti calcolati per mezzo di metodi di raccomandazione differenti siano presentati insieme. Ad esempio, mettendo insieme un metodo Collaborative e un Content-Based, otterremmo un suggerimento basato sui feedback degli utenti, e parallelamente una valutazione sulle descrizioni testuali degli item.

Il problema di questo approccio risiede nel caso in cui vengano forniti suggerimenti differenti. In letteratura vi sono due possibili soluzioni: stabilire un metodo dominante, in modo tale da favorire la sua previsione [19] oppure affiancare i due suggerimenti lasciando all’utente la scelta finale [20].

Ovviamente la modalità “Mixed” risolve il problema del ramp-up per il caso del nuovo oggetto inserito: in questo caso il metodo Collaborative non restituirà alcun risultato in quanto non sono presenti valutazioni da parte di altri utenti, ma comunque ci sarà un suggerimento per l’utente fornito dall’approccio Content-Based. Rimane però aperto il problema del nuovo utente, in quanto vengono utilizzati due metodi con apprendimento automatico che soffrono di questo problema.

1.2.3 Approccio “Features Combination”

La modalità “Feature Combination” consente di fondere due o più metodi facendo in modo che il risultato di un sistema di raccomandazione sia per l’altro una ulteriore feature per l’oggetto.

Per esempio, supponendo di voler combinare il metodo Collaborative e il Content-Based, sarà possibile utilizzare le informazioni provenienti dal sistema collaborativo come semplici feature aggiuntive associate a ciascun oggetto da gestire poi con il metodo Content-Based. Più semplicisticamente, le raccomandazioni per gli oggetti provenienti dal primo metodo serviranno come “aiuto” al secondo metodo per restituire un suggerimento.

Basu et al [21], in un sistema di raccomandazione per film, hanno coniugato il metodo Collaborative e il Content-Based ottenendo un notevole aumento della significatività delle risposte. Ciò è dovuto al fatto che la “Features Combination” fa in modo che il sistema combini i dati collaborativi senza fare affidamento esclusivamente su di essi.

1.2.4 Approccio “Cascade”

La modalità “Cascade” prevede la combinazione di due o più metodi di raccomandazione in un processo per fasi. In altre parole, il primo algoritmo di produrrà un insieme di oggetti suggeriti. Tale lista verrà raffinata dai metodi successivi fino ad arrivare ad una raccomandazione pulita e molto efficiente. Uno dei primi sistemi presenti in letteratura che presentano questo approccio è *EntreeC* [2], ovvero un sistema di raccomandazione per ristoranti.

EntreeC coniuga il metodo Knowledge-Based ed il Collaborative. In particolare, il primo è usato per scremare i possibili oggetti interessati in funzione delle richieste dell'utente (ricordiamo come il metodo Knowledge-Based funzioni in base a un questionario (o similari) riempito dall'utente e che fornisce i suggerimenti in base alle risposte). In questo modo, dopo il primo passo, la lista "grezza" viene organizzata in classi di equivalenza. Il metodo Collaborative entra in gioco in ogni classe, evidenziando quale ristorante debba essere consigliato.

L'approccio "Cascade" è molto conveniente dal punto di vista computazionale, evitando che il metodo con bassa priorità venga applicato su tutti gli oggetti dello schema. Esso si focalizza solo su quelli valutati interessanti dal primo metodo di raccomandazione. Un ulteriore vantaggio consiste in una buona tolleranza dell'errore, in quanto il metodo a bassa priorità non andrà mai a capovolgere l'esito del primo metodo ma può solo raffinarlo.

1.2.5 Approccio "Feature Argomentation"

La modalità "Feature Argomentation" sfrutta un metodo per produrre una valutazione o una classificazione di un oggetto. L'informazione prodotta è incorporata nel calcolo del metodo di raccomandazione successivo.

Uno degli esempi più noti in letteratura è *Libra*[22], che effettua raccomandazioni di libri basandosi sui dati prodotti dal sistema di raccomandazione di Amazon ® (<http://www.amazon.com>) per mezzo di un classificatore che lavora con una rete Bayesana. Il classificatore lavora sui dati come "autori correlati" e "titoli correlati" generati da Amazon per mezzo del proprio sistema Collaborative.

Questo approccio è interessante in quanto permette di migliorare le performance di un metodo (come quello di Amazon) senza dover effettuare interventi strutturali.

Si osservi che l'approccio "Feature Argomentation" è sensibilmente differente dall'approccio "Feature Combination" in cui sono combinati dati grezzi provenienti da diverse sorgenti.

Per alcuni aspetti si può accostare questo approccio con il metodo “Cascade” in quanto entrambi applicano il metodo a bassa priorità sui risultati conseguiti dal metodo ad alta priorità; tuttavia gli approcci rimangono sostanzialmente diversi perché con l’approccio “Feature Argumentation” le informazioni utilizzate dal metodo a bassa priorità includono l’output del primo metodo.

1.2.6 Approccio “Meta-Level”

L’approccio “Meta-Level” permette di combinare due metodi di raccomandazione in modo che il modello generato dal primo sia input per il secondo. Si distingue dall’approccio “Feature Argumentation” dove si usa il risultato della prima selezione per generare l’insieme delle informazioni da dare in input al secondo metodo.

Un esempio di sistema di raccomandazione “Meta-Level” è fornito da Pazzani [10] all’interno di un sistema per la raccomandazione di ristoranti. Propone un metodo ibrido Content-Based + Collaborative; in particolare, per mezzo del primo algoritmo descrive le feature con cui vuole valutare i ristoranti graditi dall’utente.

Il vantaggio primario nell’utilizzare questo approccio fondendo due metodi come quello Content Based e quello Collaborative (come nell’esempio presentato) consiste nel fatto che il modello generato dal primo metodo costituisce una rappresentazione compatta degli interessi dell’utente e tale rappresentazione risulta utilissima nella computazione del risultato da parte del metodo collaborativo, che può quindi lavorare in modo più proficuo rispetto alle semplici valutazioni dell’utente che verrebbero usate in un metodo Collaborative puro.

1.2.7 Approcci ibridi a confronto

L’approccio ibrido consente di ridurre (o di risolvere) i problemi che affliggono singolarmente i metodi puri utilizzati. Ovviamente, se si affiancano due algoritmi che soffrono del problema del ramp-up per il nuovo utente, l’approccio ibrido risultante continuerà ad avere questo problema.

Per questo motivo, indipendentemente dalla modalità o dall'ordine con cui si decide di combinare due metodi puri, è necessario sceglierli in modo tale che i punti di forza dell'uno compensino le debolezze dell'altro. I metodi Knowledge-Based e Utility-Based sono tra i più adatti a far parte di una combinazione in quanto non soffrono del ramp-up.

Sembrerebbe però che questa considerazione sia stata già smentita in quanto diversi approcci ibridi che sono stati presentati precedentemente contengono sia il metodo Collaborative sia il Content-Based: ad un'analisi attenta si evince che, sebbene entrambi siano affetti dal ramp-up, in molte delle situazioni in cui sono applicati, le informazioni sulle valutazioni degli utenti sono inferibili dal tracciamento dati dell'utente, oppure esiste un database di partenza.

Un ulteriore aspetto da considerare nella creazione di un approccio ibrido è l'ordine con cui i metodi puri sono accoppiati: in particolare, per gli approcci "Weighted", "Mixed", "Switching" e "Feature Combination", l'ordine in cui metodi coinvolti è ininfluenza; ciò non è vero per gli approcci "Cascade", "Feature Argumentation" e "Meta-Level" che sono intrinsecamente ordinati.

Ciò accade quando negli approcci non ordinati il suggerimento è calcolato indipendentemente dall'altro metodo, solo nell'ultimo passaggio i metodi "vengono a contatto" unendo i rispettivi risultati. Al contrario, negli approcci ordinati, il primo metodo fornisce una lista "grezza" di risultati al secondo, che la raffina fornendo il suggerimento corretto.

Non sempre ha senso combinare due metodi. Supponiamo per esempio di fondere il metodo Knowledge-Based con il metodo Collaborative, oppure Demographic, o ancora con un Content-Based in modalità "Feature Combination": si osserva che per definizione il metodo Knowledge-Based può prendere in considerazione qualunque tipo di dato. Per questo motivo questo tipo di approccio ibrido è assolutamente privo di senso.

Nello studio della definizione del miglior approccio ibrido è necessario valutare il comportamento di ciascun metodo all'interno dello spazio delle raccomandazioni. In particolare è possibile distinguere due casi:

- un metodo di raccomandazione ha una migliore precisione dell'altro sull'intero spazio (*uniform case*);

- i due metodi hanno un diverso grado di significatività in differenti parti dello spazio (*non-uniform case*);

Se i metodi sono uniformemente disuguali avrà senso impiegare un metodo ibrido affinché sia contenuto l'errore del metodo debole, pur mantenendo i vantaggi per i quali è stato considerato. Il discorso, ad esempio, vale se vengono utilizzati il metodo Utility-Based che è una specializzazione del metodo Knowledge-Based. A tal fine converrebbe utilizzare l'approccio "Cascade" dove l'algoritmo più forte ha maggiore priorità rispetto a quello debole, oppure il sistema "Feature Argumentation" dove il metodo debole contribuisce in una piccola parte alla formulazione del suggerimento.

Capitolo 2 – Applicazioni pratiche

Il capitolo precedente ci offre una panoramica sulle tecniche algoritmiche disponibili attualmente per la realizzazione di un sistema di raccomandazione efficiente.

I sistemi di raccomandazione hanno centinaia di applicazioni, dall'e-commerce, all'Ingegneria del Software, all'e-Learning (di cui parleremo nel dettaglio nel capitolo 3) e molte altre, ognuna con regole particolari, affinità e punti di distacco dalla raccomandazione pura e astratta di cui si è parlato in precedenza.

Le applicazioni che analizzeremo sfruttano gli approcci ibridi tentando di ridurre o di eliminare del tutto gli effetti negativi dei metodi puri che partecipano alla sua creazione. Vedremo come molti di loro utilizzano il metodo Collaborative come base di partenza, in modo da sfruttare le esperienze passate di altri utenti per la formulazione del suggerimento.

Analizziamo ora quattro applicazioni pratiche di sistemi di raccomandazione, cercando di estrapolare punti di forza, debolezze, funzionalità e come questi sistemi software possono semplificare e ridurre il carico di lavoro o la complessità del compiere una decisione.

2.1 RSSEs – Recommendation Systems for Software Engineering

Lo sviluppo del software si scontra continuamente con l'introduzione di nuove tecnologie, partendo dai nuovi componenti architettonici dei calcolatori, passando per nuovi framework o design pattern, arrivando infine a nuovi linguaggi di programmazione.

Al giorno d'oggi, i progetti di livello medio-alto sono composti da decine di migliaia di righe di codice, molteplici classi, interfacce, librerie di dimensioni spropositate. Lo sviluppatore, pertanto, deve essere in grado di sapersi orientare all'interno di questa vastità di materiale disponibile. In questo contesto nascono i sistemi di raccomandazione per l'Ingegneria del Software (RSSE) che aiutano il programmatore in molteplici attività, dalla stesura dei bug reports, al riuso di codice già scritto.

Una definizione di RSSE ci viene fornita da Martin Robillard, docente di *Computer Science* presso la McGill University, Canada: “Un RSSE è un’applicazione software che, in un dato contesto, fornisce oggetti informativi che possono essere utilizzati per un processo di ingegneria del software”.

Vista l’enorme varietà di operazioni che uno sviluppatore può compiere su un software, esiste una vastità di RSSE che tentano di aiutare il programmatore a prendere la giusta decisione. In commercio sono disponibili sistemi quali CodeBroker [23], le cui immense potenzialità vengono sfruttate per riutilizzare righe di codice scritte in precedenza, oppure Strathcona [24], che facilitano la scelta di esempi da utilizzare nella documentazione; il programmatore potrebbe aver bisogno di un suggerimento per scegliere quale metodo (o funzione) chiamare in un determinato momento (ParseWeb [25]). Altri tipi di RSSE suddividono il codice in liste organizzate e compattate in modo da facilitare la ricerca della giusta sezione (Snade [26]), a cui segue la scelta di ciò che deve essere migliorato prioritariamente (eRose [27]).

Questa diversità di aiuti che gli RSSE possono fornire al programmatore ci permettono di distinguerli in tre macro-categorie in base alla loro funzionalità primaria:

- *raccolta dati*, raccolgono dati sul processo di sviluppo creando una sorta di schema;
- *motore di raccomandazione*, analizzano gli schemi e forniscono suggerimenti;
- *interfaccia utente*, pannello di controllo con il quale l’utente chiede il suggerimento e legge la risposta del sistema.

Presenteremo adesso un RSSE per ciascuna categoria.

2.1.1 eRose

Navigando su Amazon.com ci vengono forniti suggerimenti del tipo “Altri utenti che hanno acquistato questo articolo hanno anche...”: questi suggerimenti vengono forniti mantenendo una traccia sulle vendite.

Il plug-in di Eclipse IDE chiamato eRose compie sostanzialmente la stessa operazione, estraendo le versioni precedenti dei software da archivi specifici,

come il *Current Version Systems*(CVS). Questa operazione traccia gli elementi che subiscono cambiamenti e aggiorna i suggerimenti forniti dal sistema di raccomandazione ad ogni salvataggio delle modifiche. Per esempio, se uno sviluppatore volesse aggiungere una nuova preferenza a Eclipse IDE, e quindi cambiasse *fKeys[]* e *initDefaults()*, il sistema di raccomandazione eRose suggerirebbe <Cambia *plugin.properties*> in quanto diversi altri sviluppatori che hanno aggiunto una preferenza all'IDE hanno cambiato quella proprietà.

Nel fornire un suggerimento, per prima cosa eRose elimina dalla lista delle possibili raccomandazioni gli oggetti del contesto applicativo, in quanto sono di proprietà dello sviluppatore e si suppone che, se necessitassero di modifiche, al momento dell'utilizzo del RSSE saranno già stati modificati. Successivamente, valuta gli oggetti rimanenti in base al numero di transazioni differenti a cui appartengono (più frequentemente viene utilizzato, più è stato modificato per rispondere alle richieste specifiche dello sviluppatore). Infine eRose rivela le dipendenze interne di ogni oggetto: ad esempio, se un programmatore modificasse un oggetto di un database, dovrebbe modificare anche il diagramma che rappresenta lo schema implementato.

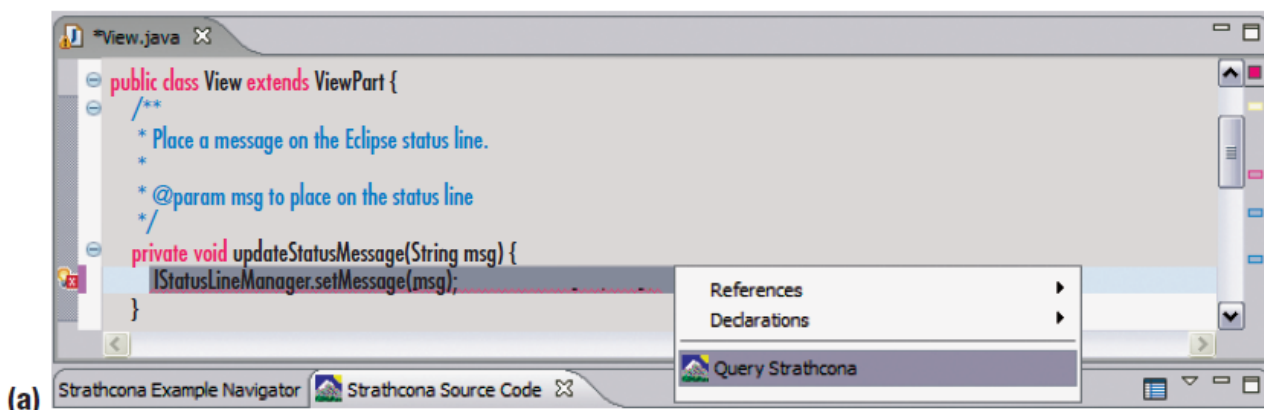
2.1.2 Strathcona

Un framework costituisce un grande insieme di esempi pratici di codice che aiutano lo sviluppatore nella fase di programmazione. Tuttavia possono essere così vasti e complessi che spesso la ricerca del codice giusto risulta proibitiva e molto onerosa in termini di tempo.

Il sistema Strathcona tenta di recuperare dal framework esempi di codice sorgente che possono aiutare il programmatore nella scrittura del codice per il suo progetto: per esempio, nell'ambiente Eclipse IDE, se lo

sviluppatore volesse modificare a livello di codice una parte della sua applicazione, potrebbe evidenziare la parte di codice che vuole modificare (figura 1a) , effettuare una chiamata a Strathcona chiedendogli un suggerimento per modificare tale parte. Strathcona cercherà tra gli esempi disponibili quelli che si adattano a quella particolare sezione di codice (che può essere una *view* su piattaforma Android, come una *status bar*) e restituirà il suggerimento.

Figura 1a – Selezione di codice da modificare, chiamata della query Strathcona



Strathcona utilizza un query PostgreSQL per navigare all'interno della *repository* in modo da estrapolare una lista di suggerimenti possibili che verranno poi scremati tramite una serie di regole euristiche. Da questa seconda fase uscirà un elenco di suggerimenti, di cui verranno mostrati solo i primi 10. Di ognuno, Strathcona mostra due formati, una sorta di diagramma a blocchi (figura 1b) e il codice effettivo di ciascun esempio (figura 1c).

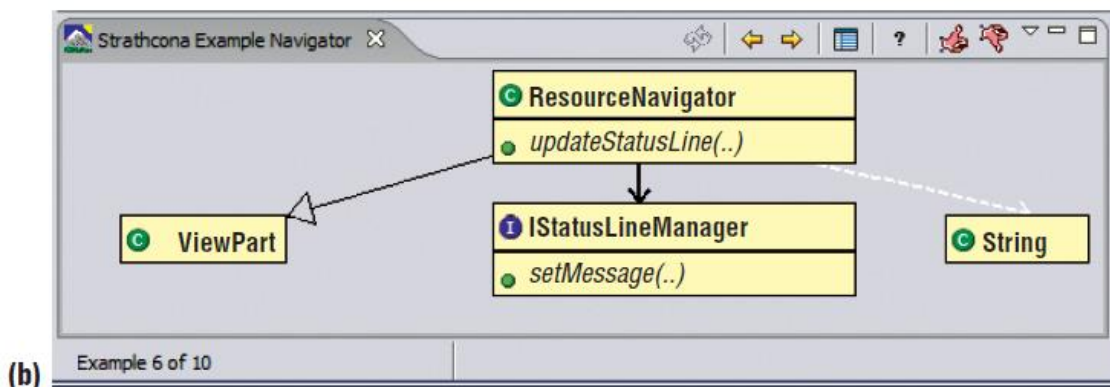


Figura 1b – Diagramma a blocchi dei suggerimenti Strathcona

```

/**
 * Updates the message shown in the status line.
 *
 * @param selection the current selection
 */
protected void updateStatusLine(IStructuredSelection selection) {
    String msg = getStatusLineMessage(selection);
    getViewSite().getActionBars().getStatusLineManager().setMessage(msg);
}
    
```

(c)

Figura 1c – Codice di programmazione dei suggerimenti forniti da Strathcona

Inoltre mostra al programmatore il fondamento logico della raccomandazione (figura 1d).

Razionale	Artifact
Class Inherits From	org.eclipse.ui.part.ViewPart
Method Calls Method	org.eclipse.jface.action.IStatusLineManager.setMessage(Ljava.lang.String;)
Method Uses Type	org.eclipse.jface.action.IStatusLineManager
Method Uses Type (S)	java.lang.String

OK Cancel

(d)

Figura 1d – Motivazione logica della raccomandazione

2.1.3 Suade

Suade è un plug-in di Eclipse IDE che genera suggerimenti automatici per la fase di controllo del codice in post-produzione.

Per ottenere un suggerimento, lo sviluppatore specifica esplicitamente una serie di caratteristiche rilevanti e metodi che andranno a formare il contesto di raccomandazione, che verranno utilizzati da Suade per trovare oggetti rilevanti alle richieste del programmatore e quindi fornire suggerimenti.

Suade ordina gli oggetti rilevanti organizzandoli in un grafo in base alle loro dipendenze statiche, applicando poi delle regole euristiche per una valutazione in ordine di importanza e rilevanza per le necessità dello sviluppatore.

I suggerimenti verranno poi presentati tramite view. Gli sviluppatori dovranno trascinare il suggerimento selezionato all'interno della finestra dove è presente il codice e le modifiche verranno apportate in automatico (figura 2).

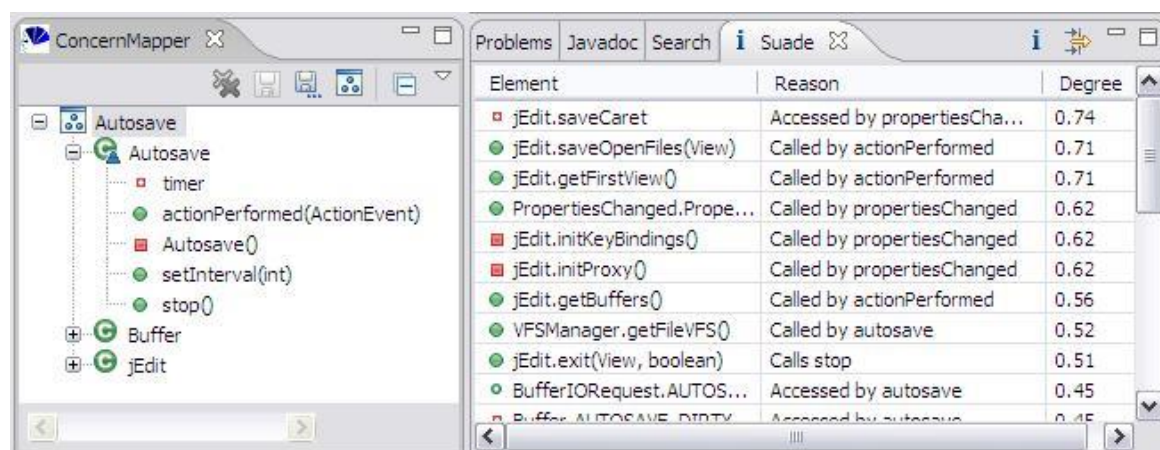


Figura 2 - Suade

2.2 ADDRESS – Asan Adverse Drug Reaction EMR Surveillance System

Al giorno d'oggi la pratica diagnostica come la Risonanza Magnetica richiede l'utilizzo di mezzi di contrasto radioattivi (RCM) per una migliore definizione e individuazione di malattie.

Sebbene questi isotopi radioattivi siano generalmente accettati dall'organismo umano, determinate persone soffrono di ipersensibilità che può causare problemi che spaziano da semplici pruriti o eritemi fino a shock anafilattici con arresti respiratori.

Questi problemi possono essere risolti iniettando nel paziente, prima del mezzo di contrasto, una certa quantità di antistaminici o corticosteroidi. L'ospedale universitario Asan Medical Center, nella Corea del Sud, ha studiato un metodo di somministrazione che prevede un sistema di raccomandazione per stabilire quale medicinale e in quale quantità somministrare al paziente.

L'ADDRESS è un sistema software in uso dal Marzo 2003 presso l'ospedale coreano, a cui è associato un database che memorizza i dati di tutti i pazienti che subiscono esami diagnostici che richiedono l'utilizzo di un mezzo di contrasto.

Dopo che un paziente subisce un DHR⁴, viene aggiunto un record all'interno del database di ADDRESS. Se in seguito lo stesso paziente dovesse necessitare di un esame diagnostico richiedente un mezzo di contrasto, all'atto della prenotazione apparirà a schermo un pop-up che avvertirà il medico prescrivente del DHR rilevato precedentemente. Il sistema avvertirà lo specialista della sostanza chimica alla quale il paziente è risultato allergico e in automatico imporrà al medico la prescrizione di una cura di antistaminici e/o corticosteroidi da effettuare prima dell'esame diagnostico per prevenire reazioni allergiche. Ovviamente le dosi da somministrare sono dettate dal medico.

Nel 2009 il sistema venne migliorato, introducendo due modifiche che semplificano il lavoro del medico nella prescrizione della cura preventiva; l'ipersensibilità viene distinta in due categorie:

- *RCM hypersensitivity reactions*: sensibilità al mezzo di contrasto senza effetti palesi e visibili;

⁴ Drug Hypersensitive Reaction, reazione allergica o ipersensibilità a un dato mezzo di contrasto

- *DHRs other than RCM hypersensitivity reactions*: dove l'ipersensibilità viene manifestata tramite eruzioni cutanee (*mild hypersensitivity*) o in casi più gravi da dispnee, broncospasmi o addirittura arresti cardio-respiratori (*severe hypersensitivity*) [28];

La seconda modifica introdotta consiste nella raccomandazione automatica della cura preventiva, in base alla reazione precedentemente riscontrata (*mild* o *severe*): non essendoci un protocollo ufficiale e standard per la somministrazione di antistaminici o corticosteroidi per prevenire reazioni allergiche a mezzi di contrasto, la cura deve essere adattata per ciascun paziente. Il sistema valuta una serie di condizioni, come altezza, peso, medicinali già assunti al tempo dell'esame diagnostico, record DHR inseriti nel database dell'ospedale e formula una lista di medicinali, con la loro quantità e modalità di assunzione.

Dalla seguente tabella (Tabella 1) si evince come ADDRESS ha ridotto considerevolmente le reazioni allergiche nella sua prima versione, per poi andare a migliorare sempre di più i risultati con le nuove migliorie introdotte nel 2009

	Previous System	Current System
<i>Demographic Data</i>		
Sex(M/F)	220/142	261/142
<i>Age (years)</i>		
<20	3	0
20-39	30	15
40-59	178	189
>60	151	189
<i>Index Reaction</i>		
Severe	48 (13%)	103 (26%)
Mild	314 (87%)	300 (74%)
<i>Breakthrough reaction (%)</i>		
Mild reaction	46 (12.7%)	24 (6%)
Severe Reaction	9 (2.5%)	3 (0.7%)

Tabella 1

Lo studio intrapreso dai medici coreani rileva l'importanza dell'uso dei sistemi di raccomandazione in ambito medico per la prevenzione di effetti collaterali dovuti all'uso di mezzi di contrasto in pratiche diagnostiche.

Al momento ADDRESS è il primo e unico sistema di raccomandazione in ambito medico attivo al mondo [28], e i dati presentati incoraggiano il suo utilizzo e la sua espansione anche in altri settori della medicina.

2.3 Sistemi di raccomandazione Firewall

Un firewall è un componente passivo di difesa perimetrale di una rete informatica, che può anche svolgere funzioni di collegamento tra due o più tronconi di rete, garantendo dunque una protezione in termini di sicurezza informatica della rete stessa. [29]

Solitamente, un firewall esamina ogni pacchetto che tenta di accedere alla rete interna, cercando di instradare correttamente il pacchetto verso la sua destinazione.

Bloccando le comunicazioni non autorizzate da/per la rete interna, i firewall proteggono contro l'attacco di hacker che potrebbero cercare di compromettere la sicurezza della rete installando applicazioni spyware per carpire dati dagli host oppure utilizzarli come "ponte" per attacchi informatici ad altri sistemi.

Definire le regole e configurare un firewall efficiente può risultare un lavoro ostico, in quanto si potrebbe ottenere un software con regole troppo semplificate che permette una notevole comunicazione o che blocchi pacchetti entranti che sono effettivamente richiesti.

La realizzazione di un sistema software che fornisca suggerimenti per un firewall produce metodi, servizi e software per raccomandare politiche di sicurezza, dove i suggerimenti sono forniti tramite un sistema Collaborative, che raccoglie i feedback di un gran numero di utenti sulla composizione dei pacchetti affidabili e non affidabili.

Esistono diverse implementazioni di questi sistemi, le quali hanno dei punti focali in comune; il procedimento di raccomandazione è articolato in 4 fasi:

1. Un pacchetto passa per il router gateway e viene analizzato dal firewall;
2. Il firewall invia una richiesta al sistema di raccomandazione per determinare se il pacchetto debba essere scartato perché non affidabile o instradato verso la giusta destinazione;
3. Il sistema di raccomandazione analizza la richiesta confrontando il pacchetto oggetto di inchiesta con un proprio database interno tentando di capire se è affidabile o meno;
4. Il sistema restituisce una valutazione in base alla quale il firewall instrada il pacchetto o lo scarta.

Per effettuare la valutazione di affidabilità, il sistema di raccomandazione ha a disposizione un database con le informazioni sulle connessioni bloccate in passato dai firewall che partecipano al metodo collaborativo. Il sistema confronta le connessioni memorizzate nel database con quella attiva al momento della richiesta di valutazione.

Viene assegnato un voto (*positive/negative*) alle comunicazioni storicizzate, dove un voto positivo indica l'affidabilità del collegamento, e viceversa. La valenza di un voto positivo alla connessione attiva viene fornita da una media ponderata delle valutazioni positive delle comunicazioni storicizzate simili ad essa. [30]

Il database memorizza informazioni raccolte da diversi firewall riguardo le comunicazioni analizzate in passato, definendo regole euristiche e algoritmi atti alla formulazione della valutazione. Il sistema cataloga le informazioni identificando le affinità tra di esse, suddividendole nuovamente tra positive e negative.

2.4 Sistemi di raccomandazione per l'e-commerce turistico

Un applicativo e-commerce per il turismo consiste nell'uso della tecnologia informatica per la produzione di informazioni, prenotazione di biglietti, hotel, viaggi in ambito turistico. [31]

Gli e-commerce turistici lavorano in due fasi: la prima prevede il marketing online e la vendita di prodotti turistici, come biglietti aerei, prenotazioni di alberghi, ecc.. La seconda fase si concentra sull'applicazione delle tecnologie informatiche per migliorare l'efficienza delle aziende turistiche e ridurre i loro costi nel fornire servizi ai clienti.

Essendoci una notevole quantità di siti web che forniscono un servizio e-commerce per il turismo (basti pensare ai più comuni booking.com, trivago.it, expedia.it), gli utenti si trovano a dover cercare di trovare il giusto compromesso tra migliaia di offerte che gli vengono mostrate, con un notevole spreco di tempo. Allo stesso tempo, le aziende che forniscono il prodotto visualizzato dai siti sopracitati sono costrette a barcamenarsi tra centinaia di tipologie di inserzione possibili per promuovere la loro offerta e renderla più visibile per gli utenti.

Un sistema di raccomandazione per l'e-commerce turistico aiuta le aziende che forniscono il servizio nel promuovere la loro offerta per soddisfare al meglio le aspettative e le richieste del cliente.

Come si può intuire, un sistema di raccomandazione per l'e-commerce turistico basato sul web ha come punto focale un database contenente gli oggetti da raccomandare. Analizziamo nel dettaglio le due fasi di implementazione del sistema di raccomandazione, la definizione del database e la sua struttura e la realizzazione vera e propria del software.

2.4.1 Web Data Mining

Un sistema di data mining sul web è formato da tre parti:

- *web content mining*: si occupa di carpire informazioni dai campi testo, immagini, audio e generalmente dagli oggetti di una pagina web. Dal punto di vista del recupero delle informazioni, questa fase aiuta l'utente

a filtrare le informazioni necessarie; dal punto di vista dell'estrazione dal database, è la fase in cui si semplifica l'utilizzo dei dati web;

- *web structure mining*: si occupa di tracciare un grafo di navigazione del sito web cercando tra le sue pagine(i nodi del grafo) link ad altre pagine(gli archi del grafo). Questa fase aiuta a trovare l'index page di un sito in modo da ottimizzare la ricerca al suo interno;
- *web usage mining*: tenta di prevedere il comportamento del sito web nell'interazione con gli utenti, spulciando i file di log del server, dei cookies. Aiuta a fornire agli utenti un servizio personalizzato in base ai loro interessi.

2.4.2 Sistemi di raccomandazione e-commerce

Un sistema di raccomandazione basato sul web data mining influenza un'applicazione e-commerce su tre aspetti. Primo, aiuta l'utente a trovare l'oggetto desiderato più velocemente e pertanto aumenta le vendite del sito e-commerce. Secondo, con l'aiuto di un sistema di raccomandazione, l'utente visualizza l'oggetto cercato, ma non sa da quale sito sta comprando, aumentando così il commercio tra siti differenti (<http://www.trivago.it>). Terzo, analizzando i file di log, i cookies e le abitudini degli utenti il sistema di raccomandazione provvede a fornire suggerimenti più precisi.

I sistemi di raccomandazione e-commerce possono essere classificati in tre tipi:

- *Non-personalized recommender systems*: il sistema suggerisce gli stessi oggetti a utenti differenti, valutandoli in base al numero di vendite registrate. Il processo di raccomandazione può terminare automaticamente;
- *User-based recommender systems*: il sistema dal principio cerca di trovare delle caratteristiche in comune tra gli utenti, creando una sorta di categorie, e poi formula il suggerimento in base agli acquisti precedenti degli utenti. Il metodo più usato per questo tipo di sistema è quello Collaborative;

- *Product-based recommender systems*: il sistema fornisce suggerimenti in base alle caratteristiche in comune degli oggetti. Il metodo più usato per questo tipo di sistema è l'Association Rule Mining.

2.4.3 Raccomandazione tramite metodo Association Rule

Il metodo di raccomandazione Association Rule venne proposto da Agrawal nel 1993 per analizzare le associazioni tra prodotti differenti nel carrello della spesa. [32] L'Association Rule Mining è un approccio molto utilizzato per scoprire cosa diversi oggetti hanno in comune: il contesto in cui è maggiormente proposto è quello dei database relazionali, in cui le associazioni possono essere rappresentate come *regole associative* o *oggetti frequenti*. L'obiettivo del metodo è quello di trovare la regola secondo la quale tutti gli oggetti sono presenti in una qualche relazione, con dei margini d'errore preimpostati.

Il processo può essere diviso in due parti:

1. Ricerca degli oggetti frequenti: assumendo un valore $MinSup=p$, gli oggetti frequenti sono quelli la cui probabilità di essere suggeriti è maggiore di p .
2. Ricerca della regola associativa: trovare la regola secondo la quale l'intervallo di confidenza degli oggetti frequenti è maggiore dell'intervallo minimo preimpostato.

2.4.4 Applicazione in ambito turistico dei sistemi di raccomandazione per e-commerce

Un e-commerce turistico offre servizi come la prenotazione di biglietti, hotel, viaggi, tour, guide turistiche e così via. Utilizzando un sistema di raccomandazione, l'applicazione web può risultare molto più efficiente e accurata.

Come descritto in precedenza, i sistemi di raccomandazione influenzano l'attività e-commerce in diversi aspetti, due dei quali sono la maggior precisione dei suggerimenti (Collaborative Filtering) e la vendita dei servizi da più fornitori tramite lo stesso e-commerce (Association Rule). Avendo già trattato il metodo

collaborativo, analizziamo passo dopo passo come funziona il metodo Association Rule.

Nell'esempio successivo, prendiamo il caso di un'agenzia di viaggi che propone 5 diverse soluzioni.

1. *Preparazione del database:*

solitamente, in questo tipo di applicazione, ci sono tre diversi database: uno contenente le informazioni degli utenti (*User Information Database*), uno che memorizza gli ordini effettuati (*Ordering Database*) e uno per i prodotti disponibili (*Product Database*) (Tabella 2):

User Information Database		Ordering Database		Product Database	
<i>Field Name</i>	<i>Type</i>	<i>Field Name</i>	<i>Type</i>	<i>Field Name</i>	<i>Type</i>
<u>Username</u>	char	<u>Username</u>	Char	<u>Product_number</u>	Char
<u>Password</u>	char	<u>Order_number</u>	Num	Product_name	cha
Gender	char	Product_name	char	Product_price	num
Age	char	Product_number	num		
Income	char	E-Mail	char		
E-Mail	char				

Tabella 2 – Legenda: Username (primary key)

2. *Riorganizzare il database:*

per poter utilizzare il metodo Association Rule, bisogna ristrutturare il database originale come mostrato dalla Tabella 3:

Product	Product's number	Products Interested	Transaction number (ID)
Product 1	1	1,3,4	0001
Product 2	2	1,2,4	0002
Product 3	3	2,3,5	0003
Product 4	4	1,2,4,5	0004
Product 5	5	1,2,3	0005

Tabella 3

3. *Ricerca dell'oggetto più frequente e formulazione del risultato:*

Il programma estrae automaticamente gli oggetti frequenti. Assumendo che $MinSup=2$ otteniamo i seguenti passaggi:

- i) Scansione del database riorganizzato al punto 2. cercando gli oggetti frequenti e calcolo del *first-level frequent item set* (Tabella4):

Product	1	2	3	4	5
Support	4	4	3	3	2
Frequent Item	Yes	Yes	Yes	Yes	Yes

Tabella 4

- ii) Calcolo del *second-level frequent item set* (Tabella 5):

Product Set	{1,2}	{1,3}	{1,4}	{1,5}	{2,3}	{2,4}	{2,5}	{3,4}	{3,5}	{4,5}
Support	3	2	3	1	2	2	2	1	1	1
Frequent Item	yes	yes	yes	no	yes	yes	yes	no	no	no

Tabella 5

- iii) Calcolo del *third-level frequent item set* (Tabella 6):

Product Set	{1,2,3}	{1,2,4}	{1,2,5}	{2,3,4}	{2,3,5}	{3,4,5}
Support	1	2	1	1	1	1
Frequent Item	no	yes	no	no	no	no

Tabella 6

Otteniamo quindi un terzo livello che ci offre una terna {1,2,4} di oggetti accomunati da una qualche caratteristica. Procedendo ipoteticamente ad un quarto livello, otterremmo zero risultati e quindi il processo termina automaticamente.

2.4.5 Realizzazione del sistema di raccomandazione

Il processo di raccomandazione basato sul web data mining è composto di due parti, *on-line module* e *off-line module*. Vediamoli nel dettaglio:

- Il modulo off-line consiste si occupa di raccogliere i dati necessari per la formulazione del suggerimento, dalla lettura dei file di log, ai cookies sul browser e così via.

- Il modulo on-line si occupa della formulazione del suggerimento. Il processo è completato dal motore di raccomandazione in base al risultato dell' metodo previsto. Quando un utente visita un sito web, il browser comunica con lui al fine di raccogliere informazioni sull'utente collegato e inviarle al motore di raccomandazione, che genererà dei suggerimenti in base alle richieste.

Capitolo 3 – Tecniche di raccomandazione e-Learning

Il presente capitolo si pone l'obiettivo di analizzare alcune tecniche per la raccomandazione in ambito e-Learning.

Trattandosi di metodi per suggerire oggetti di apprendimento (*learning object*) collocato in ambienti di formazione ICT, è necessario illustrare le due dimensioni che le tecniche prendono in esame, la dimensione tecnologica e la dimensione pedagogica. Per la prima verrà descritta brevemente la formazione a distanza (FaD), con un breve excursus su come si è evoluta nel tempo, e verranno tracciati i limiti di questa tecnologica.

Per quanto riguarda la dimensione pedagogica, verranno prese in esame le caratteristiche cognitive dello studente, molto importanti per la realizzazione di un sistema adattivo e personalizzabile in base alla capacità di apprendimento dell'utilizzatore del sistema di raccomandazione. In particolare affronteremo una discussione sugli stili cognitivi e le strategie didattiche.

Concluderemo presentando alcune tecniche presenti in letteratura per la raccomandazione in ambito e-Learning.

3.1 Il contesto tecnologico

Il sempre più esteso utilizzo del web come luogo di diffusione della conoscenza ha progressivamente favorito l'affermarsi di varie opportunità educative, portando allo sviluppo di ambienti ICT tecnologicamente avanzati per l'erogazione di corsi on-line e servizi didattici.

Ciò che oggi si chiama e-Learning nasce dall'integrazione di due diversi campi di applicazione delle tecnologie didattiche: i sistemi di apprendimento basati sull'uso del computer (*Computer Based Learning*) e la FaD. È per questo motivo che presenteremo l'analisi dei primissimi sistemi realizzati negli anni sessanta fino a giungere agli ambienti per la FaD attualmente in uso.

Lo studio del contesto tecnologico si conclude con una breve descrizione dei punti di forza e debolezza degli attuali ambienti di formazione al fine di evidenziare le linee di intervento da adottare affinché diventino strumenti strategici capaci di migliorare il processo di insegnamento-apprendimento e di

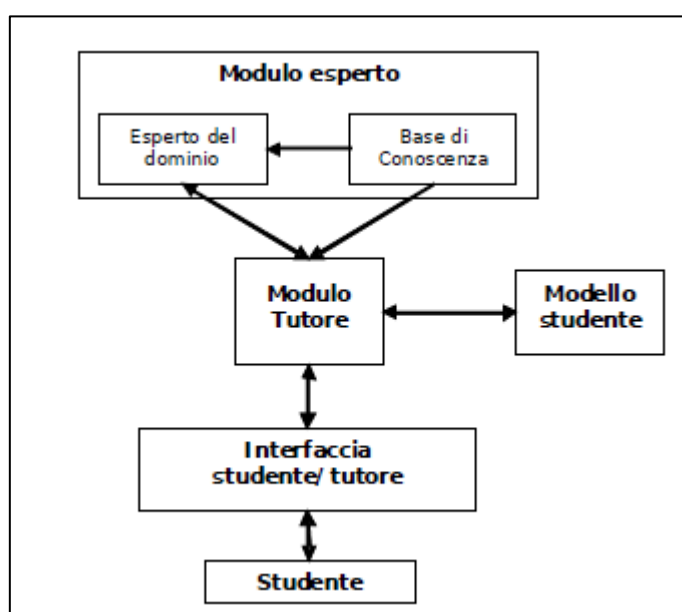
sviluppare nello studente abilità cognitive e metacognitive⁵ troppo spesso trascurate dagli attuali sistemi di formazione on-line. [33]

3.1.1 Computer Based Learning

Negli anni sessanta, con i primi studi che tentano di coniugare informatica e didattica, nascono i sistemi CAI (*Computer Aided Instruction*). Nonostante questi sistemi vengano ancora utilizzati, dal punto di vista della ricerca rivestono esclusivamente un'importanza storica.

Il materiale didattico viene partizionato in frame didattici (videate) che possono essere presentati indipendentemente o seguire un filo logico predefinito. Ogni frame viene seguito da una batteria di quesiti o esercizi con lo scopo di verificare o rinforzare le conoscenze acquisite. Non avendo particolare capacità di problem solving possono risultare troppo rigidi [34]. Gli svantaggi principali sono la struttura predefinita del materiale didattico (unica per tutti gli studenti), la fruizione passiva (lo studente non può intervenire per modificare il processo didattico) e la valutazione quantitativa e non qualitativa dello studente.

Negli anni settanta nascono gli ITS (*Intelligent Tutoring System*), che sono



frutto dell'intersezione dell'intelligenza artificiale, psicologia cognitiva e pedagogia. Come evoluzione dei sistemi CAI, gli ITS tentano di guidare lo studente attraverso il percorso didattico simulando il comportamento di un docente umano. Danno una certa libertà di azione allo studente, cercando allo stesso tempo di indirizzarlo

sulla strada corretta ogni volta che se ne discosta troppo [35]. Per far ciò, il

⁵ L'abilità metacognitiva di una persona è quella capacità che gli esseri umani hanno di riflettere sui propri stati mentali e di dirigere i propri pensieri. È una caratteristica peculiare che, tra le altre cose, permette di controllare i processi di apprendimento [Wikipedia – http://it.wikipedia.org/wiki/Teoria_della_mente]

programma deve capire le necessità dello studente e avere conoscenza sugli argomenti che lo studente deve imparare, su quelli che conosce e sulle strategie di apprendimento. L'architettura di un ITS riflette queste caratteristiche (Figura 3):

il sistema presenta, oltre all'interfaccia utente, il *modulo esperto*, il *modello studente*, il *modulo tutore*.

Il modulo esperto ha una doppia funzione, agisce come sorgente della conoscenza da presentare e valuta le performance dello studente. Deve anche generare le soluzioni

intermedie per determinare quale percorso risolutivo è seguito dallo studente.

Il modulo tutore è costituito da un insieme di specifiche relative a quale materiale deve essere presentato, in quale modalità e in quale momento. È responsabile dell'effettiva interazione con lo studente.

3.1.2 Formazione a distanza

La storia della formazione a distanza segue l'evoluzione delle tecnologie di comunicazione ed è caratterizzata essenzialmente da tre generazioni. La prima risale al 1840 in Inghilterra e consiste nell'insegnamento per corrispondenza, basato sulla distribuzione di materiale cartaceo; la seconda si basa sull'insegnamento per mezzo di prodotti multimediali, utilizzando simultaneamente materiale stampato trasmissioni televisive, audio, video e software didattico; la terza, infine, prevede l'insegnamento a distanza on-line tramite Internet.

La FaD di terza generazione ha in sé vastissime potenzialità derivanti dall'uso della rete globale e dalla sua potenza e diffusione. Vedremo come è strutturata, analizzando le piattaforme attualmente in uso, i learning object e i loro standard.

3.1.3 Piattaforme di e-Learning

Le piattaforme di e-Learning sono applicazioni web dinamiche capaci di gestire i processi di insegnamento-apprendimento a distanza. In ognuna di esse sono presenti tre macro-componenti: [36]

- *LCMS (Learning Content Management System)*;
- *LMS (Learning Management System)*;
- *Tool di Comunicazione*;

Il LCMS è il componente che gestisce i contenuti didattici. Il suo scopo è creare, importare ed esportare i contenuti. Attualmente, la maggior parte di questi strumenti si occupa semplicemente di importare file in qualsiasi formato e renderli disponibili allo studente tramite un catalogo di corsi tra i quali può scegliere.

La componente LMS, invece, include tutti i servizi necessari alla gestione delle attività per l'insegnamento on-line che vanno dalla gestione degli studenti alle attività di monitoraggio e reporting delle operazioni effettuate dagli studenti, in termini di numero di accessi al sistema, materiale didattico consultato e così via.

I tool di comunicazione hanno l'obiettivo di simulare le dinamiche di una classe reale in un ambiente virtuale.

Dallo studio sullo stato dell'arte delle piattaforme di e-Learning si evince come le aziende sviluppatrici si preoccupino maggiormente di produrre ambienti che offrano un gran numero di servizi di comunicazione e di gestione [33] mettendo in secondo piano l'aspetto didattico-pedagogico. Funzionalità di monitoraggio dell'attività degli studenti hanno un impatto molto basso quando sono di fondamentale importanza per l'adattamento del percorso per il singolo studente.

3.1.4 Learning Object

Il concetto di learning object nasce da un'idea di Reigeluth e Nelson [37] il cui intento era quello di risolvere il problema degli insegnanti di dover essere spesso costretti a scomporre il materiale recuperato dalla rete per adattarlo alle proprie esigenze di insegnamento.

L'idea base dei LO è la creazione di piccoli blocchi d'istruzione indipendenti dal contesto e autonomi che possano essere assemblati e riutilizzati in maniera diversa un numero indefinito di volte anche in contesti differenti, senza doverne riprogettare il contenuto.

Ogni nuova composizione permette di dar luogo a percorsi didattici distinti ed adeguati alle esigenze dell'utente finale a cui si rivolge l'intervento educativo.

Dal punto di vista della ricerca, un learning object può essere definito come la somma di due elementi: contenuto formativo e metadati. Il primo necessario per il trasferimento della conoscenza, il secondo utile a fornire il valore aggiunto alla risorsa di apprendimento.

3.1.4.1 Standard per i Learning Object

Un LO rappresenta un'unità minima di apprendimento per l'acquisizione di conoscenza rispetto ad un obiettivo formativo. Grazie ai metadati che lo descrivono, un LO può essere facilmente reperibile e può essere usato dall'autore o da altri utenti che lo aggiornano o lo modificano in funzione del nuovo contesto d'uso.

I metadati hanno quindi la funzione di descrivere in maniera strutturata e precisa la risorsa di apprendimento, in modo tale da permettere un efficiente reperimento sul web e un suo corretto utilizzo. I metadati sono standard, leggibili da qualsiasi sistema per poterne facilitare la ricerca. Generalmente, esistono due tipi di metadati:

- *metadati interni*, integrati alla risorsa e trasportabili con essa, per la gestione e l'implementazione del sistema;
- *metadati di comunicazione inter-sistemi*, come quelli che consentono la ricerca e l'accesso a contenuti di sapere, biblioteche virtuali e rete (standard SCORM).

3.1.4.2 Learning Object Repository

I Learning Object Repository (LOR) sono sistemi di archiviazione, ricerca e diffusione dei LO. Sono considerati lo strumento con cui le risorse didattiche passano dall'autore al fruitore e come il luogo fisico in cui vengono depositati i LO prodotti. La particolarità di un LOR consiste nel fornire un motore di ricerca

che utilizzi i metadati per consentire all'utente di individuare la risorsa didattica digitale che più rispecchia la sua esigenza [38]. Le componenti fondamentali di un LOR sono:

- database di LO e metadati;
- motore di ricerca che localizza i LO all'interno del database;
- interfaccia utente (solitamente un portale web) che consente agli utenti di caricare o scaricare LO, o di interagire con il motore di ricerca.

In merito a quest'ultimo componente, è importante far notare che i LOR hanno la capacità di comunicare tra loro. È stata implementata, infatti, una rete di Repository interoperabili, nell'ottica di condivisione delle risorse. Per questo motivo, nel 2003, la IMS⁶ ha realizzato delle specifiche internazionali per lo sviluppo e l'implementazione dei Repository chiamata *IMS Digital Repositories Interoperability Specification* (DIRS) al fine di consentire agli utenti di effettuare ricerche contemporaneamente nelle diverse banche dati.

3.1.5 Analisi del contesto tecnologico

Le piattaforme di e-Learning, così come i LO, racchiudono grandi potenzialità. Essi coniugano due dei temi più importanti della comunicazione moderna: Internet e la multimedialità.

Attualmente l'e-Learning sfrutta poco questo formidabile connubio. Le attuali piattaforme sono ancora un mero veicolo per l'erogazione dei contenuti didattici anziché uno strumento capace di elaborare conoscenza in maniera intelligente e personalizzata. Infatti:

- non permettono la personalizzazione dell'insegnamento sulle reali esigenze dello studente;
- non permettono l'adozione di modelli didattici innovativi e rimangono spesso legati al modello tradizionale di didattica frontale;
- sfruttano i risultati delle esercitazioni solo per la reportistica sui progressi dello studente e non per adattare il percorso formativo;

⁶ *IMS Global Learning Consortium*, organizzazione non profit con l'obiettivo di produrre criteri di interoperatività e riutilizzo delle piattaforme di e-learning.

- non riescono a valutare autonomamente parametri pedagogici relativi ai singoli studenti essenziali per l’ottimizzazione del processo di formazione;
- non offrono supporto intelligente ai docenti nella creazione dei corsi;
- non offrono supporto intelligente agli studenti nella scelta degli obiettivi formativi.

3.2 Il contesto pedagogico

Un intervento formativo accurato, se è importante nell’apprendimento in presenza, diventa fondamentale nell’apprendimento a distanza.

Se in una classe reale il docente può rendersi conto con un colpo d’occhio se gli studenti sono attenti, cosa accade in una classe virtuale? Come può l’apprendimento on-line essere utile al discente?

È necessario che il sistema informatico conosca come ciascun individuo si approccia all’apprendimento ed in funzione di ciò valuti la migliore strategia di insegnamento da adottare. Affinché l’e-Learning sia uno strumento veramente efficace, è necessario motivare l’utente offrendo una formazione che coniughi le sue esigenze cognitive (ciò che deve imparare) con le preferenze metacognitive (il modo migliore per impararle).

Dal momento che ogni individuo ha preferenze metacognitive diverse, un sistema di apprendimento a distanza deve avere la capacità di adattarsi al singolo studente, considerando tutte le sue caratteristiche. Per questo motivo è necessario analizzare le teorie pedagogiche presenti in letteratura, ed in particolare analizzeremo i contenuti formativi e gli stili cognitivi, per comprendere come lo studente può essere descritto mettendo in evidenza gli aspetti più utili alla personalizzazione del percorso formativo.

La scelta del contenuto formativo più adatto al profilo dell’utente richiede l’analisi di aspetti differenti, relativi al modo in cui è fornito l’intervento didattico. Affinché si possa suggerire il LO più adatto in base alle caratteristiche dello studente è necessario considerare tre aspetti: l’obiettivo didattico, la forma del contenuto e la strategia d’insegnamento.

3.2.1 Obiettivi didattici

L'obiettivo didattico definisce il risultato che lo studente può raggiungere in termini qualitativi, ovvero di abilità che è possibile acquisire utilizzando il learning object.

La conoscenza qualitativa può essere caratterizzata come:

- *dichiarativa*, intesa come l'insieme dei concetti, delle definizioni e dei principi teorici che è possibile trasferire. Per mezzo della conoscenza dichiarativa non si impara ad applicare i concetti appresi: per esempio, in matematica, lo studio della definizione di limite non spiega sempre come calcolarne uno
- *procedurale*, traduzione di uno specifico sapere appreso con la conoscenza dichiarativa in azioni adeguate al conseguimento di uno scopo. Prendendo sempre l'esempio della matematica, lo studio di funzioni è conoscenza procedurale;
- *metacognitiva*, intesa come capacità logica, creativa, critica. Lo sviluppo delle abilità metacognitive è alla base dell'apprendimento.

3.2.2 Forma del contenuto

La forma del contenuto è una discriminante fondamentale nel processo di insegnamento-apprendimento. Essa indica il modo con cui il contenuto è trasferito dall'insegnante allo studente.

Gli aspetti da considerare per caratterizzarla sono tre:

- *forma esteriore*, il modo in cui il contenuto si manifesta fisicamente (video, testo, simulazione);
- *struttura interna*, che indica le parti che compongono il contenuto (concetti, procedure, connessioni, osservazioni, ...);
- *tipologia di conoscenza*, indica il modo in cui il contenuto è veicolato. Può essere dichiarativo se descrive in modo statico la struttura dell'oggetto, procedurale se spiega il modo per raggiungere un obiettivo fornendo un insieme di regole.

Se, ad esempio, la forma esteriore è di tipo testuale e la conoscenza di tipo dichiarativo, il testo sarà descrittivo se enuncia le caratteristiche di un oggetto,

espositivo se informa su un fatto accaduto, argomentativo se la conoscenza è utilizzata per riflettere, discutere o ragionare sul concetto.

Nel caso in cui la forma esteriore sia di tipo testuale e la conoscenza di tipo procedurale, il testo risultante sarà di tipo regolativo in quanto indicherà azioni da eseguire in presenza di condizioni specifiche.

La scelta del formato del contenuto più appropriato alle caratteristiche dello studente deve essere effettuata in base al dominio e al contenuto da trasmettere e secondo le preferenze dello studente.

3.2.3 Strategie di insegnamento

La strategia di insegnamento da adottare è di fondamentale importanza per la formulazione del suggerimento da parte del sistema di raccomandazione. Esse indicano il tipo di intervento formativo che l'educatore applica allo studente per facilitargli l'apprendimento.

Le strategie di insegnamento sono numerose ma possono essere suddivise in due categorie:

- *attive*, implicano un coinvolgimento diretto dello studente che partecipa in prima persona all'esperienza educativa, parlando, svolgendo esercizi;
- *passive*, si limitano all'ascolto, lettura o alla visione di materiale multimediale (come ad esempio le classiche lezioni frontali insegnante-classe).

Una strategia passiva si divide in lezione e modalità tutoriale. La lezione comporta l'ascolto dei concetti da parte dello studente, trasmessi per mezzo della voce del professore o slides multimediali. La modalità tutoriale si basa sulla divisione dell'insieme degli argomenti da trasmettere allo studente in unità distinte, orientate al raggiungimento di uno specifico obiettivo accompagnate da prove di valutazione per verificare l'apprendimento di ogni singola unità.

Fra le strategie attive si annoverano:

- *modellamento*, osservazione e imitazione da parte dello studente della procedura mostrata dal sistema che permette di portare a termine un certo compito;
- *discussione*, scambio-confronto di idee tra docente e studenti;

- *caso di studio*, presentazione di un problema reale da analizzare, su cui gli studenti formuleranno ipotesi di risoluzione. Lo studente deve utilizzare le teorie apprese in precedenza per trovare la soluzione. Può essere molto utile per colmare il gap tra teoria e pratica;
- *simulazione*, riproduzione di problemi ed eventi quanto più verosimili, consentendo però allo studente di agire in un ambiente controllato e di apprendere dai propri errori. La simulazione si presenta in vari modi, dal gioco alla realtà virtuale;
- *role playing*, simulazione di una situazione in cui gli studenti si identificano con specifici personaggi di una scena drammatica, assumendone il ruolo e comportandosi di conseguenza;
- *problem solving*, soluzione di problemi che consente agli allievi di rivivere i processi tipici dei ricercatori (ipotesi-verifica).

La scelta della strategia didattica dipende da fattori come il tipo di conoscenza da trasmettere, oppure dal contenuto didattico che lo studente deve apprendere.

3.2.4 Stili cognitivi – Modello di Felder

Personalizzare un'azione formativa significa scegliere il materiale più idoneo alle esigenze dello studente sia dal punto di vista degli obiettivi sia per le strategie di insegnamento. Ma quali sono i parametri che determinano tale scelta?

Possono essere oggettivi, come il tipo di conoscenza da acquisire, oppure soggettivi, come le preferenze cognitive dello studente. Per questo è molto importante, per la creazione del profilo utente, determinare lo stile cognitivo con il quale lo studente apprende le nozioni.

Sostanzialmente esistono quattro scuole di pensiero per l'identificazione dello stile cognitivo dello studente. Analizzeremo solamente la più importante, dalla quale si è preso spunto per la formazione del profilo utente per un sistema di raccomandazione generico, ossia il modello chiamato ILS (*Index of Learning Style*).

Secondo Felder, autore di questo modello, gli studenti hanno differenti modalità e preferenze nel modo di acquisire nuova conoscenza. In collaborazione con Linda Silverman [44] ha definito un modello per classificare gli allievi in base alle loro caratteristiche di apprendimento. Uno studente può essere:

- *percettivo-intuitivo*, per cui si prende spunto dalle teorie psicologiche di Carl Jung [45] e dal tipo indicatore di modello di Myer-Briggs [46], per cui gli studenti percettivi sono coloro che preferiscono informazioni giunte dall'esterno per mezzo dei sensi, mentre gli intuitivi privilegiano informazioni derivate dalla memoria, riflessione e dall'interpretazione.

Gli studenti percettivi tendono ad essere pratici, tolleranti con il dettaglio, bravi a memorizzare. Adottano metodi standard per la soluzione dei problemi e nutrono antipatia per le sorprese.

Gli studenti intuitivi possono essere annoiati dal dettaglio ma molto abili ad afferrare nuovi concetti e si trovano a proprio agio con astrazioni e formule matematiche. Adatti al cambiamento, fantasiosi e maldisposti a compiti ripetitivi.

Per insegnare a studenti percettivi è opportuno utilizzare esempi pratici, mentre per gli intuitivi è necessario fornire interpretazioni teoriche;

- *visivo/verbale*, che differiscono nel modo in cui ricevono le informazioni. Gli studenti visivi preferiscono rappresentazioni visuali, come immagini o video, del materiale presentato: acquisiscono meglio da grafici, schemi, fotografie e difficilmente assimilano ciò che gli viene spiegato oralmente. Gli studenti verbali privilegiano informazioni trasmesse per via orale e ricordano meglio quel che sentono rispetto a quel che vedono. La maggior parte degli studenti viene classificata come visiva.

Un insegnamento efficace per gli studenti visivi richiede la preparazione di materiale didattico in cui siano evidenziati i punti chiave e in cui caratteristiche simili vengano evidenziate nella stessa maniera (con colori uguali). Per gli studenti verbali è preferibile presentare le informazioni oralmente o con diapositive testuali.

I processi mentali responsabili dell'elaborazione dell'informazione ci permettono di distinguere le persone in due categorie:

- *sperimentatori attivi*, che memorizzano le informazioni applicandole. È preferibile che le nozioni vengano somministrate per mezzo di discussioni o applicazioni per mezzo di esercizi. Per questi studenti si preferisce un approccio collaborativo o apprendimento tra pari. Per insegnare a uno studente attivo l'istruttore deve alternare una breve discussione con attività di problem solving;
- *osservatori riflessivi*, sono validi teorici, modellatori e matematici. Preferiscono definire i problemi e proporre le possibili soluzioni; necessitano di tempo per maneggiare mentalmente le informazioni e dedurre il significato. Per insegnare a uno studente riflessivo è necessario alternare fasi in cui si presenta la teoria a momenti dedicati alla riflessione.

La quarta dimensione in cui si suddividono i discenti è quella sequenziale-globale:

- *studente sequenziale*, assorbe le informazioni e preferiscono studiare il materiale secondo un procedimento logicamente ordinato, costruendo la conoscenza a piccoli passi. Questi studenti seguono processi di “ragionamento lineare” per risolvere i problemi e sono generalmente ordinati nella formulazione di risposte e soluzioni;
- *studente globale*, impara frammenti di conoscenza apparentemente disgiunti, lasciando trascorrere del tempo fino a che improvvisamente intuiscono il concetto globale. Hanno un pensiero divergente e spiccate capacità di sintesi. Esigono tempo per la comprensione fino a che non afferrano il quadro di insieme e scorgono i collegamenti con altri argomenti in modo più perspicace rispetto agli studenti sequenziali.

3.3 Sistema di raccomandazione e modello studente

Come abbiamo visto nei precedenti esempi nel capitolo 2, attualmente le tecniche di raccomandazione rimangono legate a campi come l'e-commerce o il news filtering e esistono pochi esempi di applicazioni per e-Learning. Anche dal punto di vista dei metodi applicati, si può notare come sia l'approccio

Collaborative il più utilizzato, sostanzialmente per la sua adattabilità essendo indipendente dal dominio applicativo.

Nell'e-Learning, invece, il dominio applicativo è di fondamentale importanza in quanto per essere veramente funzionale all'apprendimento dello studente, il materiale suggerito deve esser adatto e posto in maniera consona per ogni singolo utente che usufruisce del sistema di raccomandazione, e come abbiamo visto in precedenza, ogni studente ha singole necessità dettate dalle proprie caratteristiche cognitive.

Il problema principale di un sistema di raccomandazione per e-Learning è la realizzazione del modello utente. Nei primissimi sistemi educativi la somministrazione del materiale era del tutto indipendente dal tipo di studente che lo riceveva. Questa cosa è andata migliorando nel tempo, e in seguito, con gli ITS, i sistemi sono diventati consapevoli della conoscenza che veicolano e di quanto velocemente viene appresa dallo studente. Ciò è possibile grazie al modello studente che memorizza le sue abilità, i suoi obiettivi formativi e tutto quando gravita attorno al processo di insegnamento-apprendimento. I sistemi ITS sono capaci di prendere in considerazione le istanze relative allo studente e al contesto di apprendimento al fine di guidarlo lungo tutto il percorso formativo.

Le piattaforme di e-Learning hanno un ampissimo insieme di dati sullo studente, possono esaminare i log, esercizi svolti e risultati ottenuti. Sono un ottimo strumento per la raccolta del feedback implicito. Mancano tuttavia della conoscenza più importante per l'adattamento dell'azione formativa: la conoscenza del dominio.

3.3.1 Rappresentazione della conoscenza del dominio

Lo studio sui metodi di modellazione degli aspetti cognitivi nasce con i primi sistemi ITS.

Il primo modello in ordine di tempo è quello *overlay* proposto da Carr e Goldstein [47] ce modella la conoscenza dello studente come un sottoinsieme di quella posseduta dal sistema, che vengono quindi confrontate e questo permette al sistema di capire cosa lo studente deve imparare. Lo svantaggio

dell'utilizzare questo modello sta nel fatto che lo studente non può possedere conoscenze diverse da quelle del sistema. per tale motivo, il modello prevede che tutte le differenze riscontrate dal sistema tra il suo comportamento e quello dello studente siano imputabili alla mancanza di una o più abilità dello studente. Inoltre, nel caso in cui l'utente presentasse una strategia risolutiva diversa da quella memorizzata nel sistema, questa non sarà riconosciuta e sarà valutata errata.

Successivamente venne definito il modello *buggy*, detto anche "differenziale", che permette la rappresentazione di errori e fraintendimenti comuni fatti dal soggetto [48]. Questo modello ammette la possibilità che lo studente oltre alla conoscenza corretta, possa averne una errata. Durante l'interazione, quando il sistema rileva un errore nella conoscenza dello studente, ne identifica la causa facendo riferimento ad una libreria di errori che viene modificata in itinere, facendo riferimento ad errori comuni di altri studenti tramite un sistema collaborativo. L'errore viene memorizzato nel modello studente, la cui capacità dipende direttamente dalla qualità dell'implementazione della libreria di errori.

Il modello a *perturbazione* è un ibrido tra questi due metodi [49]. Rappresenta la conoscenza dello studente come un sottoinsieme della conoscenza del sistema (overlay) includendo anche le possibili conoscenze errate (*buggy*). Inizialmente è codificata esclusivamente la conoscenza del sistema; in un secondo momento viene arricchita con gli eventuali errori o incomprensioni dello studente, creando un elenco di regole o mal-regole. Sulla base dell'osservazione e dello studio del comportamento dello studente le mal-regole vengono utilizzate per creare la libreria di errori che rappresenta le perturbazioni del modello esperto. L'aggiornamento continuo di tale libreria permette un migliore rimedio agli errori dello studente, e permette di sapere se lo studente possiede conoscenze errate.

Un altro approccio alla modellazione dello studente è quella basata su vincoli, detta anche *Constraint-Based Modeling*, introdotta da Ohlsson [50]. Il punto focale prevede che le informazioni utili dal punto di vista pedagogico non siano rinvenibili dalla sequenza delle azioni dello studente, ma dalla risultante situazione a cui lo studente è arrivato [51]. Lo scopo della modellazione su vincoli è quello di superare il problema dell'ampiezza dello spazio di

conoscenza da rappresentare per mezzo di un meccanismo di astrazione realizzato con i vincoli di stato, che rappresentano i principi base del dominio di conoscenza. La notazione formale di Ohlsson per i vincoli di stato prevede una coppia ordinata (Cr , Cs) dove il primo elemento rappresenta la condizione di rilevanza che identifica la classe degli stati del problema per la quale è posto il vincolo, mentre il secondo è la condizione di soddisfazione che identifica la classe per cui il vincolo è soddisfatto. Ogni vincolo può essere pensato come un insieme delle caratteristiche o delle proprietà di uno stato del problema. In altre parole, se Cr è soddisfatto in uno stato del problema, tale stato deve soddisfare anche Cs affinché sia corretto.

Nel modello studente non sono presentate le azioni ma piuttosto gli effetti delle sue azioni. È quindi dato dall'insieme dei vincoli che sono stati violati; occorre però precisare che il fatto che un vincolo non sia stato violato non significa che lo studente abbia appreso quell'argomento.

Quanto detto in questo paragrafo sugli stili cognitivi non deve essere relegato esclusivamente ad una particolare tecnica di raccomandazione, ma deve servire come base per qualsiasi metodo sviluppato per l'ambito e-Learning.

Analizziamo ora nel dettaglio tre tecniche di raccomandazione per sistemi e-Learning presenti in letteratura, la raccomandazione Cascade, il metodo Rule-based e Moodle. Infine vedremo come un sistema di raccomandazione può essere utilizzato anche per definire gli obiettivi di uno studente.

3.4 Metodo di raccomandazione Cascade

La tecnica di raccomandazione per sistemi e-Learning proposta da Bunclé, Anane e Nakayama [53] prevede la combinazione dei metodi Collaborative e Content-based in un approccio ibrido Cascade.

La loro idea nasce dall'osservazione di una classe reale: l'interazione tra gli studenti, durante lavori di gruppo, appare come una chiara sfaccettatura del processo di raccomandazione di un metodo Collaborative, in quanto gli studenti si scambiano informazioni partecipando attivamente alla stesura della soluzione del problema a loro assegnatogli.

Nella tecnica Cascade proposta, il processo collaborativo determina affinità tra gli studenti, creando dei *neighbourh*(vicinato, gruppo), basandosi su un approccio personalizzato e su uno generale; nel primo, un profilo *implicito* è formulato per ciascun utente, sulla base delle interazioni con altri studenti e sui commenti forniti ai vari learning object di cui ha preso visione, mentre il profilo *esplicito* viene fornito dalla compilazione di un questionario al momento della registrazione, le cui risposte aiuteranno il sistema a determinare le categorie di appartenenza secondo il modello di Felder.

L'approccio generale consiste nell'utilizzare i dati forniti dalla community di studenti riguardo l'utilizzo dei vari LO e dalla loro valutazione da parte degli insegnanti.

La fusione di queste due strategie totalmente differenti nasce dalla necessità di superare il problema della *cold start* nei sistemi Collaborative, e allo stesso tempo fornisce un supporto non irrilevante agli studenti.

Entriamo ora nel dettaglio del funzionamento del sistema. Per semplicità utilizzeremo le parole *neighbourh* e gruppo in modo intercambiabile.

3.4.1 Sistema di raccomandazione LMS

Il sistema preso in esame è definito LMS (*Learning Management System*) e riesce a raggiungere contemporaneamente cinque obiettivi:

- presentazione di raccomandazioni personalizzate per ciascuno studente;

- possibilità di upload per gli insegnanti di materiale sempre nuovo ed aggiornato;
- possibilità di monitoraggio dei progressi dello studente;
- assegnamento di insegnanti/tutor per un numero variabile di corsi/gruppi;
- creazione di corsi da parte degli insegnanti per un particolare studente o gruppo;

Il sistema è stato implementato in un'applicazione web, che si appoggia su un motore di ricerca e su un database che funge da LOR⁷. Il diagramma seguente indica come vengono utilizzati i processi personalizzati e quelli globali (Figura 4).

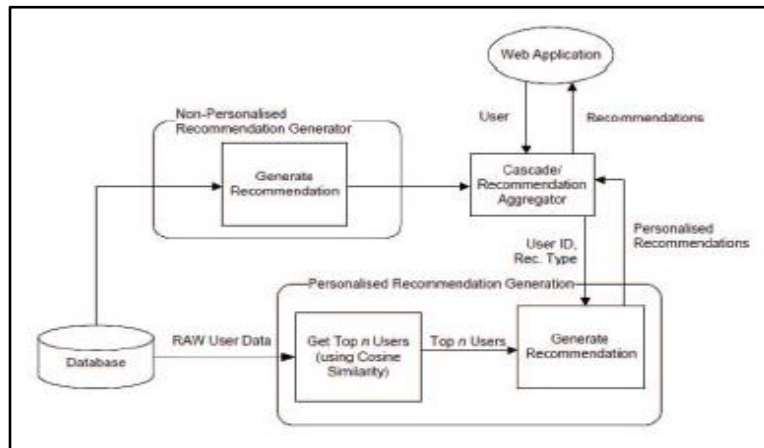


Figura 4 – Architettura del sistema

⁷ Learning Object Repository, paragrafo 3.1.4.2

3.4.2 Processo di raccomandazione

Il primo punto del processo di raccomandazione consiste nell'individuazione delle affinità tra gli studenti per la successiva divisione in *neighbourhood*. Il *Content Usage Pattern* e il *Content Rating* sono le due tecniche utilizzate per questa fase.

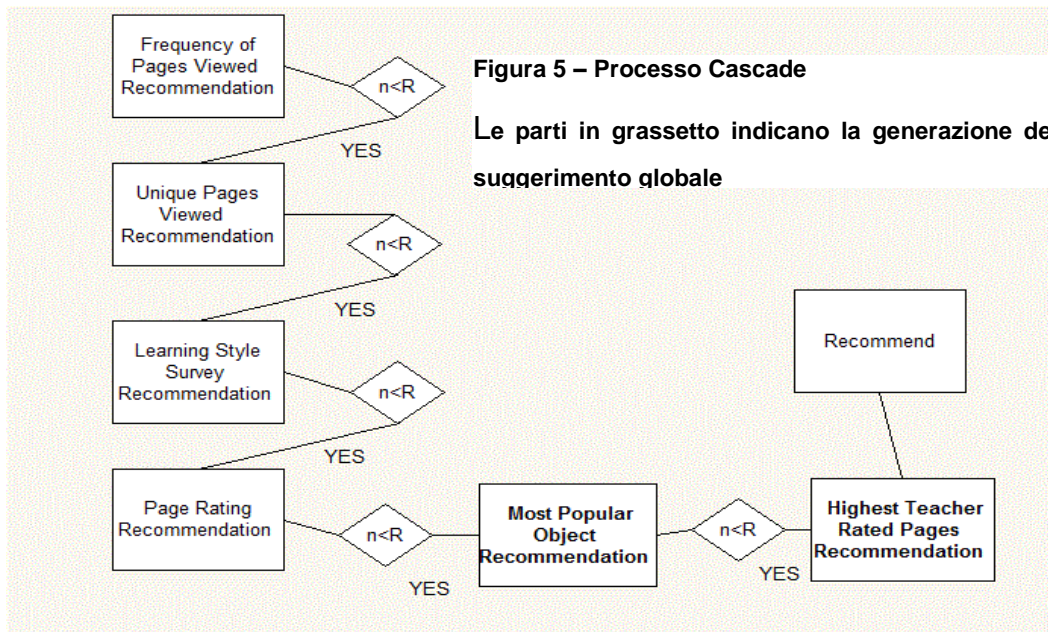
Il Content Usage coinvolge la cronologia dei LO visualizzati dagli studenti, mentre la valutazione degli oggetti (Content Rating) viene fornita esplicitamente dagli utenti. Ciò forma la base della formazione dei neighbourhood.

Si presenta ora il problema del *cold start*, in quanto se un utente si è appena registrato e non ha visualizzato alcun LO, non potrebbe far parte di alcun gruppo. Possiamo agilmente superare il problema introducendo altre tecniche per la formazione dei gruppi, basandosi su questionari proposti allo studente per determinare in maniera esplicita lo stile cognitivo e metacognitivo per inserirlo adeguatamente.

Il processo di raccomandazione a cascata prevede la formazione di gruppi per ogni fase, operando contemporaneamente suggerimenti personalizzati e globali per tutti gli studenti.

Ad ogni livello i gruppi sono generati incrementalmente, tentando di evitare che gli studenti appartengano a più gruppi. L'ultimo stadio comporta la fusione dei gruppi precedentemente formati.

Nella cascata, la generazione di ogni gruppo è valutata dalla condizione $n < R$ dove n rappresenta il numero di suggerimento generati, R il numero di suggerimenti minimi richiesti. Finché la condizione risulta vera, il processo di raccomandazione continua (Figura 5).



La generazione del suggerimento personalizzato è data dalla disamina del profilo implicito ed esplicito.

Per quanto riguarda il profilo implicito, il gruppo è generato considerando le pagine (LO) visualizzate dagli studenti. Il primo step genera coppie di utenti considerando la frequenza con la quale accedono alle pagine. Se il numero di visualizzazioni non è sufficiente, il raggruppamento viene effettuato secondo una valutazione booleana (viste/non viste) generando gruppi più ampi ma ugualmente rilevanti.

Se questa fase non raggiunge lo scopo, il gruppo viene creato esplicitamente suddividendo gli studenti in base alle loro caratteristiche cognitive e metacognitive e in seguito, se necessario, il gruppo viene ulteriormente diviso unendo gli studenti che hanno assegnato voti simili agli stessi LO che hanno utilizzato.

Se il livello della personalizzazione del suggerimento non è soddisfacente (ossia si rileva il fatto che non si sono formati gruppi), il processo a cascata passa nella fase della formulazione della raccomandazione globale.

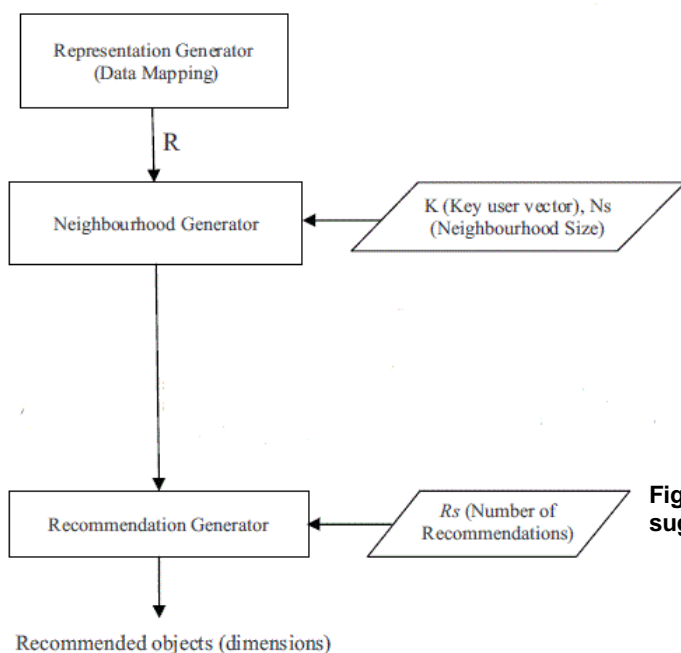
Se anche questa fase fallisce, i gruppi vengono creati con l'intervento fisico del docente che suddivide gli studenti manualmente.

3.4.3 Formulazione del suggerimento

Entrando maggiormente nello specifico, i gruppi sono determinati secondo la regola del coefficiente di Pearson e una tecnica statistica definita la *Cosine Rule*. La seguente formula è usata per calcolare la correlazione tra studenti:

$$Corr(A, U) = \frac{\sum_{i=1}^n (A_i - \bar{A})(U_i - \bar{U})}{\sqrt{\sum_{i=1}^n (A_i - \bar{A})^2} \sqrt{\sum_{i=1}^n (U_i - \bar{U})^2}} \quad (1)$$

La figura 6 mostra un chiaro esempio di come il suggerimento viene formulato. R rappresenta una lista di vettori, in cui ciascun vettore corrisponde ad un utente e ogni cella di ogni singolo vettore indica l'interazione con un LO.



La *Cosine Rule* (2), invece, richiede in input due vettori e calcola il coseno dell'angolo che, posti i due vettori come rette in un piano cartesiano, viene formato tra di essi.

Figura 6 – Processo di generazione del suggerimento

$$Similarity = \cos(\theta) = \frac{A \times B}{|A| \times |B|} = \frac{\sum_{i=1}^n (A_i \times B_i)}{\sqrt{\sum_{i=1}^n (A_i^2)} \sqrt{\sum_{i=1}^n (B_i^2)}} \quad (2)$$

3.3.4 Precisione nella formulazione del suggerimento

Dagli esperimenti condotti per valutare la tecnica si evincono ottimi risultati, espressi in termini di precisione (*Precision*) e di rilevanza del suggerimento (*Recall*).

La *Precision* viene calcolata in percentuale sulla base dei suggerimenti corretti nell'insieme dei suggerimenti complessivi forniti dal sistema. Misura, pertanto, le performance del sistema nel non effettuare suggerimenti errati.

La *Recall* rappresenta in percentuale della correttezza del suggerimento rispetto al numero massimo di raccomandazioni corrette.

$$Precision = \frac{|hit\ set|}{|n^{\circ}\ of\ recommendation|} = \frac{|H|}{|R|} \quad (3)$$

$$Recall = \frac{|hit\ set|}{|test\ set|} = \frac{|H|}{|Ts|} \quad (4)$$

Solitamente questi due valori vengono combinati in un singolo parametro, chiamato *F1-measure*:

$$F1 = \frac{2 \times recall \times precision}{recall + precision} \quad (5)$$

Possiamo vedere in figura 7 come entrambe le formule di correlazione utilizzate diano risultati ottimali

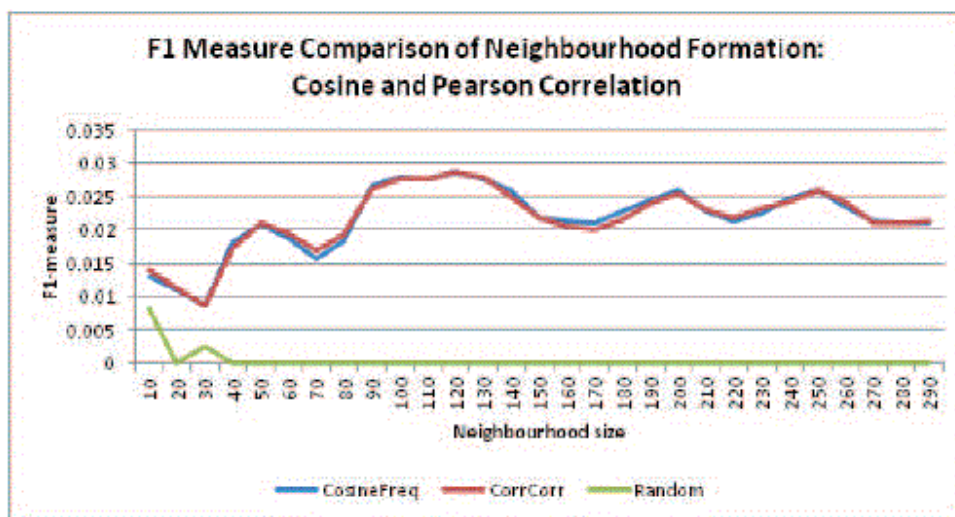


Figura 7 – Risultati raccomandazione test con coefficiente di Pearson e Cosine Rule

3.5 Tecnica Rule-based per la raccomandazione in ambito e-Learning

La tecnica proposta da Di Bitonto [52] coniuga in modalità Cascade due tecniche Knowledge-based: la tecnica Ontology-based (che sarà la primaria, ad alta priorità) che ha il compito di ricercare i LO richiesti esplicitamente e quelli ad essi collegati secondo relazioni definite nel dominio, e la tecnica Rule-based (a bassa priorità) che opera sulle caratteristiche cognitive e metacognitive dello studente per scremare i LO risultati della prima selezione che meglio rispondono a tali caratteristiche.

La tecnica ad alta priorità presenta i LO trovati organizzati in due macrocategorie: *ArgomentoRichiesto* e *ArgomentiCorrelati*. Alla prima categoria vi appartengono i learning object richiesti esplicitamente dallo studente, che riguardano l'ambito e la disciplina che il discente sta affrontando al momento dell'utilizzo del sistema. Nella seconda categoria vengono riportati i LO che la tecnica ha ritenuto opportuno collegare, che hanno una qualche affinità con la richiesta esplicita dello studente (Figura 8).

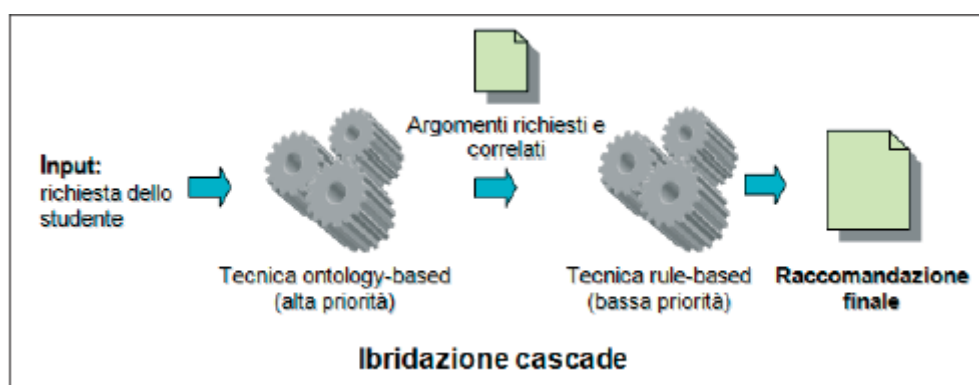


Figura 8 – Raccomandazione con tecnica ibrida

Per esempio, se uno studente richiede LO riguardanti equazioni di secondo grado, la tecnica Knowledge-based inserirà nella prima categoria quelli relativi all'argomento specifico, mentre nella categoria *ArgomentiCorrelati* inserirà LO relativi a equazioni di primo grado, interpretazione geometrica delle parabole e così via.

Avendo già affrontato nel testo la tecnica Knowledge-based e la struttura specifica dei LO, passiamo ad analizzare il funzionamento del metodo a bassa priorità Rule-based.

3.5.1 La tecnica Rule-based

Come già detto, la tecnica ad alta priorità seleziona e divide i LO in due macrocategorie. Successivamente, la tecnica Rule-based a bassa priorità stabilisce l'ordine in cui saranno presentati allo studente. Il primo argomento che lo studente visualizzerà sarà sempre quello esplicitamente chiesto dallo studente, a cui seguirà la lista degli argomenti correlati.

Per ogni argomento, un insieme di regole valuta ciascun LO in funzione dei valori dei descrittori *Argomento*, *Livello di conoscenza*, *Errori tipici* (del profilo studente) e *Obiettivo Didattico* (del LO).

Al termine della valutazione il sistema assegna un peso all'oggetto che rappresenta la sua conformità alle conoscenze dello studente rispetto al dominio. Maggiore è il peso, maggiore sarà l'adeguatezza. In funzione del peso, ciascun LO sarà classificato come *Adeguato* o *Non Adeguato* allo studente che utilizza il sistema. La somma dei pesi dei LO per ciascun argomento inciderà sulla posizione dello stesso nella lista visualizzata allo studente al termine del processo di raccomandazione.

All'interno di ogni argomento sono presentati prima i LO classificati come *Adeguato* e poi gli altri. In generale, comunque, nella stesura della lista vengono privilegiati i LO le cui caratteristiche di accesso (se sono visuali o testuali, ecc..) permetteranno allo studente di fruirne appieno, in base alle proprie caratteristiche cognitive e metacognitive.

In particolare, la posizione del singolo LO deriva dal valore della seguente combinazione lineare:

$$Pos(i) = a_1\alpha + a_2\beta + \dots + a_7\gamma + a_8\delta \quad (6)$$

Dove α , ..., δ rappresentano i valori profilo, su scala 0-5, degli stili di apprendimento dello studente (es: percettivo = 3, intuitivo = 1, ecc..), mentre i valori a_1 , ..., a_n che variano nell'intervallo [0,1] corrispondono ai valori delle componenti cognitive della strategia di insegnamento per cui il LO è stato realizzato.

Riportandoci all'esempio precedente sulle equazioni di secondo grado, la tecnica Ontology-based seleziona un insieme di LO per l'argomento "Equazioni

<pre> <output> <argomentoRichiesto nome = "Equazioni di secondo grado" > <LO id = "01" nome = "Il Delta"> <LO id = "05" nome = "Risolvere le equazioni di secondo grado"> ... <LO id = "33" nome = "Equazioni di secondo grado"> ... </LO> ... </argomentoRichiesto > <argomentoCorrelato nome = "Equazioni di primo grado" > <LO id = "54" nome = "Equazioni di primo grado"> ... </argomentoCorrelato > <argomentoCorrelato nome = Int. geom. delle equazioni di secondo grado" > <LO id = "76" nome = "Delta e parabole"> ... </argomentoCorrelato > ... </output> </pre>	<pre> <output> <argomentoRichiesto nome = "Equazioni di secondo grado" > <LO id = "05" nome = "Risolvere le equazioni di secondo grado"> <LO id = "33" nome = "Equazioni di secondo grado"> ... <LO id = "01" nome = "Il Delta"> ... </LO> ... </argomentoRichiesto > <argomentoCorrelato nome = "Equazioni di primo grado" > <LO id = "54" nome = "Equazioni di primo grado"> ... </argomentoCorrelato > <argomentoCorrelato nome = Int. geom. delle equazioni di secondo grado" > <LO id = "76" nome = "Delta e parabole"> ... </argomentoCorrelato > ... </output> </pre>
--	--

Figura 9 – Selezione dei LO tramite tecnica Ontology-based (sinistra); Rappresentazione della lista ArgomentoCorrelato (destra)

secondo grado" e altri LO per argomenti correlati come "Equazioni primo grado" o "Interpretazione geometrica eq secondo grado" (Figura 5).

La tecnica Rule-based ordina gli argomenti e i LO ad essi relativi come segue: il primo è quello richiesto, ossia "Equazioni di secondo grado" e per ogni LO su di esso, la tecnica ne calcola l'adeguatezza rispetto alle conoscenze dello studente sul dominio.

Consideriamo, ad esempio, il LO che tratta l'argomento "Delta", e supponiamo che la conoscenza dello studente sia pari a 5 per la teoria, ma solo 2 per la pratica e che non commetta errori tipici sull'argomento. Supponiamo, inoltre, che un LO sia dichiarativo, ovvero che tratti conoscenza teorica sul delta, trascurando gli aspetti legati al calcolo.

Poiché il LO veicola conoscenza già posseduta dallo studente, a questo sarà assegnato un peso molto basso, ciò significa che è poco adeguato allo studente.

Il processo continua per ogni LO di quell'argomento e produce una lista di oggetti consigliati e sconsigliati. A ciascuno di essi viene assegnato una categoria e un peso, la cui somma determina la posizione dell'argomento nella lista di raccomandazione. Supponiamo di considerare 3 LO: "Il Delta",

“Equazioni di secondo grado” e “Risolvere le equazioni di secondo grado”. Se il LO “Il Delta” è sconsigliato mentre gli altri due vengono valutati appropriati, sarà posto come terzo nella lista.

L’ordine di apparizione dei consigliati è determinato sulla base dello stile di apprendimento dello studente. In una rappresentazione vettoriale dei LO e degli stili cognitivi di uno studente, avremmo questi valori:

Vettore dello stile di apprendimento dello studente							
Percettivo	Intuitivo	Visivo	Verbale	Applicativo	Riflessivo	Sequenziale	Global
0	2	1	5	3	0	3	1

Vettore della strategia di insegnamento LO “Equazioni di secondo grado”							
Percettivo	Intuitivo	Visivo	Verbale	Applicativo	Riflessivo	Sequenziale	Global
0.2	0.7	1	0	0	0.4	1	0

Vettore della strategia di insegnamento LO “Risolvere le equazioni di secondo grado”							
Percettivo	Intuitivo	Visivo	Verbale	Applicativo	Riflessivo	Sequenziale	Global
0.7	0.2	0.3	1	0.4	0	1	0.2

Tabella 7 – Rappresentazione vettoriale modello studente – strategia di insegnamento LO

La combinazione lineare dei valori del vettore relativo allo studente con i valori del LO “Equazioni di secondo grado” da come risultato 5406, mentre la combinazione lineare dei valori del LO “Risolvere equazioni di secondo grado” da come risultato 10.1.

Questo significa che il LO “Risolvere le equazioni di secondo grado” sarà presentato prima di “Equazioni di secondo grado” come mostrato in figura 9 (lato destro).

Il processo di raccomandazione appena presentato è in grado di considerare il contesto pedagogico di riferimento sia in termini di obiettivo e modalità dell’azione educativa da attuare sia in termini di caratteristiche cognitive dello studente.

3.6 Moodle

Le attività di interazione tra persone, come chat, forum, blog, file sharing e altre stanno diventando molto popolari nell'ambiente dell'istruzione in quanto si rivelano essere strumenti che possono migliorare l'attività di apprendimento quanto l'insegnamento vero e proprio. In particolare, permettono l'implementazione di impostazioni comuni nell'apprendimento a distanza, in quanto rendono palesi i miglioramenti effettivi raggiunti dallo studente. Allo stesso tempo, sebbene queste attività favoriscano la comunicazione tra studenti, che è alla base di un sistema di apprendimento a distanza adattivo, non è semplice per l'utente scegliere le informazioni più interessanti e più utili per il suo percorso formativo.

Una piattaforma di e-Learning dovrebbe essere strutturata in modo tale da gestire in modo corretto questo sovraccarico di informazioni, cercando di identificare quale attività compiuta dallo studente può essere importante per la formulazione del suggerimento, tenendo sempre presente le abilità cognitive e metacognitive dell'utente.

Al giorno d'oggi diversi sistemi di raccomandazione sono stati sviluppati per suggerire corsi, materiale [38] o al massimo post su forum, blog presenti in rete [39] ma solo pochi di essi hanno combinato queste capacità con tecniche di suggerimento per attività di interazione per favorire la collaborazione nell'apprendimento [40].

Presenteremo adesso un approccio di raccomandazione in una piattaforma di e-Learning open-source [41], Moodle, progettata e implementata per suggerire allo studente una o più particolari attività di interazione che lo possono aiutare nel suo percorso formativo.

3.6.1 Approccio di base

Solitamente, un sistema di raccomandazione è utilizzato per suggerire ad un utente un insieme di oggetti che possono in quale modo riguardarlo o interessarlo. Il processo che porta alla formulazione del suggerimento inizia con la specifica delle caratteristiche che questi oggetti devono avere per far parte del risultato.

Gli algoritmi di raccomandazione che seguono il metodo Collaborative sono i più implementati, e sono presenti con una percentuale abbondante anche in ambito e-Learning, in quanto stimano l'utilità di una risorsa di apprendimento in base al voto attribuitole dagli utenti che l'hanno utilizzata.

Sebbene sia chiaro che i sistemi basati sul metodo Collaborative siano i più efficienti in ambito e-Learning, è stato provato che se combinati con un approccio AI possano fornire predizioni più accurate. In particolare, un metodo basato sul confronto tra voti dati da utenti, avvantaggia gli stessi (siano essi studenti o insegnanti) in termini di suggerimenti di LO, percorsi formativi o attività interattive.

3.6.2 Dynamic Clustering

Come già detto, lo scopo principale di Moodle è di suggerire attività di interazione tra utenti in modo tale da favorire l'apprendimento. In particolare, l'approccio definito nel precedente paragrafo combina il *Dynamic Clustering* degli utenti, ossia la loro divisione in gruppi omogenei in base ai percorsi seguiti e alle loro capacità cognitive, con il suggerimento dell'attività da svolgere.

Uno degli algoritmi più usati nel clustering è il *K-Means* [42] che partiziona gli oggetti dati in un numero predefinito di gruppi. Nell'e-Learning ci sono moltissime variabili da considerare, come gli obiettivi dello studente, preferenze, conoscenza a priori, e come si può facilmente capire, queste variabili cambiano velocemente nel tempo.

Per determinare il numero di gruppi da formare viene utilizzato l'algoritmo chiamato *Silhouette index*[43]. Il risultato di questo processo viene passato come parametro al K-Means che dividerà gli oggetti.

Consideriamo un insieme di n oggetti da dividere in k gruppi:

$$SII(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (7)$$

dove $a(i)$ è la varianza dell'elemento i rispetto agli altri dello stesso gruppo, $b(i)$ è la varianza dell'elemento i rispetto agli altri gruppi. La funzione $SII(i)$ fornisce

un risultato $[-1, 1]$ dove più è vicino a 1 e più l'elemento i è nel gruppo appropriato.

In questo dominio applicativo, gli studenti corrispondono agli oggetti, e ciascuno di essi viene rappresentato da un vettore che descrive i percorsi di apprendimento, capacità cognitive ecc. Utilizzando la formula (3) in una procedura iterativa, vengono calcolati diversi valori di Silhouette index corrispondenti a diversi modi di raggruppare gli studenti. Il valore scelto sarà il maggiore di tutti quelli calcolati, che fungerà da input per il K-Means, che classificherà gli studenti nel dato numero di gruppi.

Quando il processo di clustering termina, avremo k gruppi di studenti tutti con le stesse necessità di apprendimento. In ogni gruppo ci sarà uno studente "centrale" chiamato *centroide* che verrà preso in esame al momento della formulazione del suggerimento. Si assume quindi che ciò che interessa o è utile al centroide sia utile anche per gli altri studenti dello stesso gruppo.

3.6.3 Implementazione di Moodle

Dopo queste premesse, passiamo all'implementazione vera e propria del plugin Moodle. Esso è costituito da tre parti:

- *Data Gathering Component* (DGC) che estrae e analizza i dati dal Moodle DB. Come già detto, ogni studente viene rappresentato da un vettore contenente le informazioni secondo le quali verranno divisi in gruppi. Il piano di studi viene calcolato in base alle presenze a lezione e ai risultati dei test. Gli interessi dello studente vengono raccolti basandosi sulla sua partecipazione a discussioni su forum, blog. In particolare, per i forum, è calcolata la percentuale di post fatti dall'utente su uno specifico argomento, rispetto al numero totale di post fatti dallo stesso utente. Per le chat, l'interesse dell'utente viene calcolato come rapporto delle discussioni a cui ha preso parte per uno specifico argomento con le discussioni totali dell'utente.
- *Clustering Component* (CC) raggruppa gli utenti utilizzando i dati raccolti dal DGC. Prima utilizza il Silhouette index per definire il numero di gruppi, poi con il K-Means li classifica.

- *Predicate Component (PC)* calcola un punteggio per tutte le attività di interazione a cui l'utente non ha partecipato. Successivamente, calcola il coefficiente di correlazione di Pearson (formula (1)) tra l'utente e gli altri centroidi per fornire il suggerimento sulle attività di interazione non ancora svolte.

3.7 Sistemi di raccomandazione per la scelta di obiettivi didattici

Un'applicazione pratica dei sistemi di raccomandazione in ambito e-Learning consiste nel poter suggerire ad uno studente quale obiettivo didattico perseguire.

Partendo da questo concetto, Capuano et al. hanno ideato e sviluppato una piattaforma di e-Learning capace non solo di essere adattiva in base alle caratteristiche metacognitive dello studente, ma inoltre di personalizzare l'attività didattica sulla base di concetti selezionati da un modello ontologico⁸ [58].

La definizione degli obiettivi didattici viene fornita dal confronto del profilo utente dello studente con altri profili accomunati da determinate caratteristiche.

3.7.1 Intelligent Web Teacher

La tecnica di raccomandazione sviluppata da Capuano et al. è basata sull'utilizzo di una piattaforma di apprendimento chiamata IWT (*Intelligent Web Teacher*): essa offre la possibilità di creare esperienze di apprendimento totalmente personalizzabili ed è definita a partire da quattro concetti:

- *modello del dominio*: descrive la conoscenza che è oggetto di apprendimento attraverso un insieme di concetti e determina le relazioni tra di loro; una serie di *preferenze di insegnamento* possono essere aggiunte come parametro al modello del dominio;

⁸ In informatica si utilizza il termine *ontologia* per definire il modo in cui diversi schemi vengono combinati in una struttura dati contenente tutte le entità rilevanti e le loro relazioni in un dominio [[http://it.wikipedia.org/wiki/Ontologia_\(informatica\)](http://it.wikipedia.org/wiki/Ontologia_(informatica))]

- *modello utente*: rappresenta lo studente ed è costituito da uno *stato cognitivo* che misura la conoscenza che l'utente raggiunge in un momento e da un insieme di *preferenze di apprendimento* che indicano la strategia di insegnamento da adottare con lo studente;
- *modello di risorsa di apprendimento*: metadato che rappresenta un LO ed è basato sugli standard del learning object IEEE;
- *modello di unità di apprendimento*: rappresenta una sequenza di risorse che lo studente deve utilizzare per raggiungere la conoscenza prevista.

Il processo per generare un'unità di apprendimento comincia da una serie di obiettivi e da un modello utente: dapprima vengono generati una serie di domini applicativi in grado di portare lo studente al raggiungimento della conoscenza richiesta. In seguito vengono rimossi quelli contenuti argomenti già appresi dallo studente tenendo presente anche il profilo cognitivo dello stesso. Eventualmente vengono aggiunte altre unità di apprendimento che meglio si adattano al profilo dello studente.

Una delle motivazioni per cui è stato scelto IWT è la sua semplicità con la quale lo studente si può interfacciare: la piattaforma implementa le *Upper Level Learning Goals* (ULLG) che consistono in una serie di concetti del dominio con annessa descrizione testuale; possono essere forniti sia per gli studenti sia per i docenti a cui si accede tramite un motore di ricerca.

Lo studente può richiedere le unità di apprendimento in linguaggio naturale, e automaticamente il sistema restituisce le ULLG confrontando la richiesta con le descrizioni testuali associate all'interno del database.

3.7.2 Descrizione del processo

L'algoritmo proposto si occupa di fornire un suggerimento basato sul confronto tra stati cognitivi: stessi profili riceveranno medesimi suggerimenti. Per ottenere questo risultato viene utilizzato un approccio *user-to-user* collaborativo.

L'algoritmo procede in tre passi: mappatura dei concetti, valutazione dell'utilità del concetto, valutazione degli ULLG. La ULLG con il punteggio più alto sarà quella suggerita allo studente.

Sia C un insieme di concetti e L un insieme di utenti. Lo *stato cognitivo* dello studente $l \in L$ descrive la conoscenza raggiunta da l ed è rappresentata dalla funzione $CS_i : C \rightarrow [0,10]$. In sostanza, per ogni concetto c viene assegnato un punteggio da 1 a 10 in base alla qualità dell'apprendimento dello studente per quel concetto. Se il punteggio è maggiore di θ , l'argomento è definito come "appreso" altrimenti viene considerato "non appreso"

In un certo momento del suo periodo di apprendimento, ad uno studente potrebbero essere stati assegnate più unità di apprendimento. Definiamo $LPath = (c_1, \dots, c_n)$ come l'insieme delle unità possedute dallo studente in via di studio.

Partendo da questi concetti base, definiamo COT_l tutti i concetti che sono in fase di studio dallo studente l come l'unione di tutte le $LPath$ a cui lo studente è iscritto. Infine definiamo la funzione booleana *concept mapping function* $CMF: L \times C \rightarrow \{0,1\}$ descritta come:

$$CMF(l, c) = \begin{cases} 1 & \text{se } CS_i(c) > \theta \text{ o } c \in COT_l \\ 0 & \text{altrimenti} \end{cases} \quad (8)$$

Pertanto la funzione restituirà 1 se il concetto c è stato appreso dallo studente l , 0 altrimenti.

3.7.3 Valutazione dell'utilità del concetto

La valutazione dell'utilità $u(l, c)$ di un concetto c per uno studente l viene calcolata partendo dalla mappatura vista in precedenza. Partiamo dal presupposto ovvio che se un concetto è stato appreso da uno studente, esso è inutile. Pertanto se $CMF(l, c) = 1 \Rightarrow u(l, c) = 0$. Per calcolare l'utilità degli altri viene utilizzato un algoritmo di raccomandazione collaborativo.

Possiamo stimare l'utilità di c per l sommando i voti di c dati da studenti con profili simili a l , utilizzando la seguente formula:

$$u(l, c) = \frac{\sum_{l' \in L'} CMF(l', c) sim(l, l')}{\sum_{l' \in L'} |sim(l, l')} \quad (9)$$

dove L' è l'insieme degli studenti con profili simili a l e $sim(l, l')$ è il valore di somiglianza tra l e l' ottenuta utilizzando la Cosine rule (2) o il coefficiente di similarità di Pearson (1).

3.7.4 Definizione degli ULLGs

Possiamo definire formalmente un ULLG come una tupla $ULLG_i = (D_i, TC_{i1}, \dots, TC_{in})$ dove D_i è il testo associato che descrive l'unità di apprendimento in linguaggio naturale e l'insieme TC_{i1}, \dots, TC_{in} corrisponde alla lista di concetti che devono essere appresi dallo studente per raggiungere un determinato obiettivo.

IWT è in grado di generare autonomamente una $LPath$ partendo da una serie di obiettivi prefissati. Applicando il proprio algoritmo (che non tratteremo in questa sezione) è possibile determinare, per ogni $ULLG_i$ la corrispondente $LPath_i$ partendo dalla lista degli oggetti collegati tra loro.

Dopo aver determinato le liste di apprendimento associate ad un ULLG, viene stimata la *aggregated utility* $au(l, ULLG_i)$ per ogni studente l con la seguente formula:

$$au(l, ULLG_i) = \sum_{c \in LPath} \frac{u(l, c)}{|LPath_i|} \quad (10)$$

Il calcolo restituisce l'utilità degli oggetti forniti dallo ULLG per un determinato studente. Ciò implica che se l'obiettivo contiene delle nozioni già apprese, verrà restituito un valore basso anche se ci sono diversi oggetti ancora da apprendere. Pertanto è necessario introdurre il concetto di *marginal utility* $mu(l, ULLG_i)$ di un ULLG per uno studente l per limitare questo errore.

$$mu(l, ULLG_i) = \frac{\sum_{c \in LPath_i} u(l,c)(1-CMF(l,c))}{\sum_{c \in LPath_i} (1-CMF(l,c))} \quad (11)$$

Il suggerimento viene poi formulato a partire da queste due formule descritte, combinandole in una terza formula così definita:

$$u(l, ULLG_i) = \alpha * au(l, ULLG_i) + (1 - \alpha)mu(l, ULLG_i) \quad (12)$$

dove α corrisponde a un numero reale $\alpha \in [0, 1]$ scelto empiricamente per dare maggior peso all'*aggregated utility* o alla *marginal utility*, in base al risultato che si vuole ottenere.

Il suggerimento che verrà fornito allo studente sarà lo ULLG che otterrà il risultato più alto dalla (12).

Capitolo 4 - Privacy

Come possiamo evincere dai precedenti esempi proposti, i sistemi di raccomandazione sfruttano dati sensibili degli utenti, senza preoccuparsi minimamente di questioni legate alla privacy.

La privacy è il diritto alla riservatezza delle informazioni personali e della propria vita privata [54]. Di crescente rilievo è il tema della sicurezza informatica che riguarda sia i privati cittadini, sia le imprese: esso coinvolge tutti gli aspetti che riguardano la protezione dei dati sensibili archiviati digitalmente ma in particolare è noto al grande pubblico con riferimento all'utilizzo di Internet.

In effetti, la rete è in grado di offrire una vasta gamma di informazioni e servizi ma contemporaneamente può costituire un luogo pericoloso per la nostra privacy anche perché il mezzo stesso non è stato concepito per scambiare o gestire dati sensibili.

Come abbiamo notato nei precedenti esempi proposti di sistemi di raccomandazione, non sempre ci si prefigge come obiettivo quello di salvaguardare la riservatezza degli utenti del sistema. L'esempio più banale è quello di un sistema con approccio Knowledge-Based, in cui, all'atto della registrazione, l'utente deve compilare un questionario che servirà per il processo di raccomandazione; nessuna specifica di questo approccio garantisce la riservatezza dei dati inseriti, in quanto sarebbe possibile che il proprietario del sistema abbia accesso ai dati e possa venderli a terzi.

Analizziamo adesso *Pistis*, un sistema di raccomandazione per i Social Network che garantisce la riservatezza dei dati di un sistema con approccio ibrido utilizzando un metodo Collaborative e uno Content-based [55].

4.1 Pistis

I sistemi di raccomandazione Content-Based sono tra i più utilizzati nel web. Come abbiamo già visto nel primo capitolo, l'idea di base di questo approccio consiste nel raccogliere informazioni di preferenza di ciascun utente e per mezzo delle quali raggrupparli per suggerire oggetti comuni.

È triviale notare quanto facilmente possano sorgere problemi di privacy in questo approccio, in quanto le informazioni personali alla base del processo di raccomandazione vengono raccolte in unico server. In molti hanno studiato questo problema, definito in letteratura come PPCF (*Privacy-Preserving Collaborative Filtering*) [56] [57], ma le nuove problematiche sorte con l'avvento dei Social Network, dove gli utenti palesano i propri interessi commentando o condividendo oggetti hanno reso questi studi obsoleti.

Prendiamo l'esempio di Facebook ®: per creare un profilo sono necessari pochissimi minuti, e chiunque potrebbe inserire informazioni false con l'unico scopo di carpire dati sensibili di altre persone e utilizzarli per fini poco leciti.

Da quando sono nati i sistemi di raccomandazione collaborativi, non è mai stata fatta distinzione tra i dati effettivamente necessari al processo e quelli privati, inutili ai fini della formulazione del suggerimento. Per esempio, un utente dovrebbe avere la possibilità di scegliere quali dati lasciar utilizzare al sistema e quali dovrebbero essere nascosti: in un sistema classico, poniamo il caso che l'utente A commenti un articolo su un viaggio e uno su un crimine; l'utente B che commenta lo stesso articolo del viaggio di A riceverebbe un suggerimento sull'articolo criminale di A. Se dessimo la possibilità ad A di nascondere il suo "interesse" per il secondo articolo, B non riceverebbe un suggerimento errato e magari poco piacevole.

Da questo ragionamento nasce Pistis: con questo sistema, tutti i dati personali degli utenti sono memorizzati sui loro stessi computer, eliminando così il problema di un server centrale. Inoltre, gli oggetti sono raggruppati in modo tale da eliminare i collegamenti tra interessi privati e pubblici dei singoli utenti.

La valutazione dei contenuti viene eseguita tramite un calcolo multi-party che prevede che diverse processi generino il proprio output nascondendo

all'esterno l'input preso. La formulazione del suggerimento, invece, è generata a partire da argomenti di interesse raccolti sul PC dell'utente.

4.1.1 Definizione del problema

Definiamo due tipologie di interessi, pubblici e privati.

- $I_u(\textit{public}) = \{i_1, i_2, \dots, i_n\}$ per un utente u è un insieme di oggetti tali che $i_n \in I_u(\textit{public})$ se è stato pubblicato, postato o commentato da u .
- $I_u(\textit{private}) = \{j_1, j_2, \dots, j_n\}$ per un utente u è un insieme di oggetti tali che
 - l'utente ha letto j_n ;
 - $j_n \notin I_u(\textit{public})$;
 - $\forall i \in I_u(\textit{public})$, i e j non appartengono alla stessa categoria di interesse;

Indicheremo con $I_u(\textit{public})$ le informazioni che l'utente decide esplicitamente di rendere pubbliche, mentre con $I_u(\textit{private})$ quelle che vengono visualizzate ma non si vuole far vedere ad altri. In altre parole, con questa notazione, riusciamo a suddividere gli interessi personali in *pubblici* e *privati*. L'obiettivo di Pistis è quello di suggerire oggetti preservando la privacy degli utenti. Con questa suddivisione, gli interessi pubblici vengono gestiti da un classico sistema collaborativo, mentre quelli privati richiedono maggiore attenzione.

In letteratura sono presenti molte tecniche studiate per salvaguardare la sicurezza dei dati privati degli utenti, ma nessuna considera le informazioni dei Socials, oppure presentano comunque il problema del singolo server centrale che le immagazzina, che può portare alla situazione nella quale terze parti che non dovrebbero avere accesso a tali dati possano impossessarsene. Durante lo studio condotto da Dongsheng et al. nella creazione di Pistis viene ipotizzato un attacco a un Social Network cinese con più di 63k utenti, con 20k post 180k visualizzazioni giornalieri. Analizziamolo nel dettaglio.

4.1.2 Privacy Attack Model

Il Social Network preso in esame per questo test ha un sistema di raccomandazione associato e consente quattro differenti operazioni:

- creare un account;
- postare o commentare un oggetto;
- leggere un oggetto postato o commentato da un altro utente;
- richiedere un suggerimento;

Sia U l'insieme degli utenti e I l'insieme degli oggetti presenti sul social:

Definiamo il PAM come una terna $M(u, v) = \{I_u(\text{public}), I_u(\text{private}), R_v\}$ dove $u \in U$ è l'utente obiettivo dell'attacco e $v \in U$ è un profilo falso creato per condurre l'attacco. $R_v \subseteq I$ è l'insieme degli oggetti che verranno suggeriti a v .

L'utente v conduce un attacco in tre step:

1. v identifica e visualizza tutti gli oggetti presenti in $I_u(\text{public})$;
2. v richiede un suggerimento R_v e legge solamente i primi oggetti inseriti nella lista prodotta (suggerimenti più affidabili) degli elementi in cui ha partecipato (postato o commentato) u ;
3. ripete lo step 2 numerose volte e gli oggetti presenti nell'insieme $R_v - I_u(\text{public})$ forniscono un'approssimazione abbastanza corretta di $I_u(\text{private})$.

Questo modello di attacco venne testato su un Social Network cinese (Fudan BBS⁹) e gli effetti sono strabilianti.

Il modello di attacco tenta di indovinare gli interessi privati di un utente cercando di ingannare il sistema facendosi suggerire oggetti che in realtà dovrebbero essere suggeriti a u .

Per ogni utente u , indichiamo con $I_u(\text{private})$ l'insieme degli oggetti obiettivo di chi conduce l'attacco.

⁹ <http://bbs.fudan.edu.cn>

l'attaccante tenta di indovinare un insieme di oggetti $I_u(attack)$ il più vicino a $I_u(private)$ possibile. Per valutare la precisione dell'attacco, definiamo

$$AttackPrecision = \frac{|I_u(attack) \cap I_u(private)|}{|I_u(attack)|} \quad (13)$$

$$AttackRecall = \frac{|I_u(attack) \cap I_u(private)|}{|I_u(private)|} \quad (14)$$

La (13) si riferisce alla quantità in percentuale di oggetti correttamente identificati da chi conduce l'attacco.

La (14) si riferisce alla quantità in percentuale di oggetti esposti dal server che produce la raccomandazione e che vengono identificati da chi conduce l'attacco. Un valore alto di ciascuno di questi elementi indica una violazione della privacy.

La figura 10 ci mostra come questa tipologia di attacco individua con un minimo errore gli oggetti appartenenti a $I_u(private)$, ed inoltre è interessante notare come l'attacco possa risultare addirittura più efficiente del processo di raccomandazione, evento dovuto al fatto che il modo in cui viene portato crea un correlazione molto forte con la vittima.

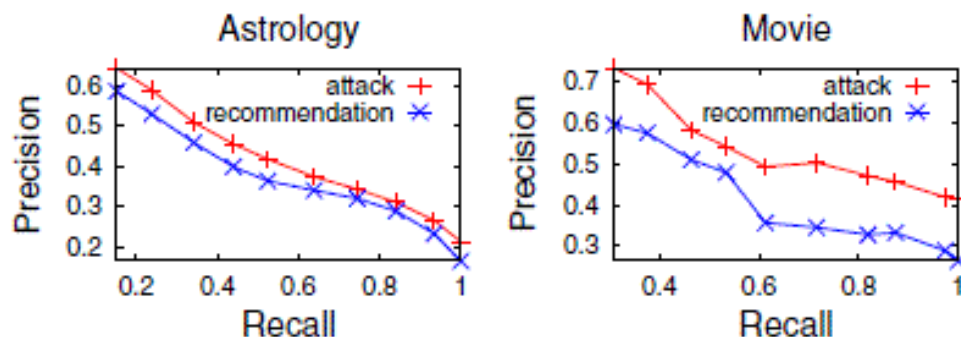


Figura 10 – Misura della precisione dell'attacco

4.1.3 Funzionamento di Pistis

La tecnica dietro questo approccio si basa su due componenti chiave:

- *identificazione di gruppi di utenti*: raggruppa utenti nello stesso modo in cui vengono raggruppati gli oggetti prima del processo di raccomandazione. Ogni utente può appartenere a più gruppi, ciascuno identificato da un interesse comune tra i suoi membri. Il protocollo di calcolo multi-party garantisce il rispetto della privacy tra i vari utenti (figura 11 – 1);
- *raccomandazione su gruppi di utenti*: utilizzando i gruppi di interesse creati nel primo passaggio, la seconda componente determina ogni punteggio di ciascun oggetto sulla base delle valutazioni pubbliche di ciascun gruppo, proteggendo i dati privati dei singoli utenti. La valutazione degli oggetti viene poi ulteriormente raffinata su ogni client personale del singolo utente, in modo da permettere un suggerimento più accurato basandosi anche sui dati privati (figura 11-2, 11-3).

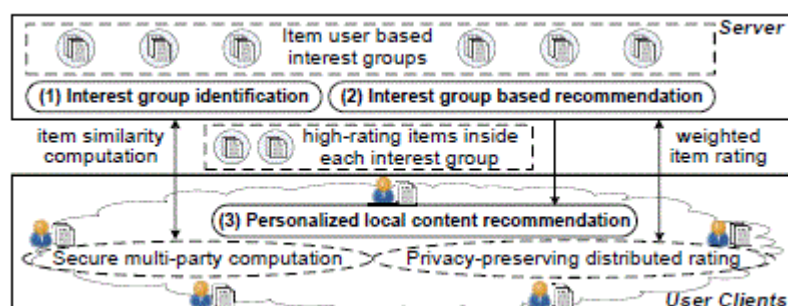


Figura 11 – Componenti chiave di Pistis

Per salvaguardare la privacy degli utenti, la cosa più intelligente da fare è troncane il legame tra gli interessi pubblici e gli interessi privati di ciascun utente.

Consideriamo l'insieme $G = \{g_1, g_2, \dots, g_k\}$ l'insieme dei gruppi di interesse creati, dove k corrisponde al numero identificativo di ciascun gruppo, che ha le seguenti proprietà:

- ogni $g \in G$ è una terna $g = \{I_g, U_g, c_g\}$ dove $I_g = \{i_1, i_2, \dots, i_n\}$ è un insieme di oggetti; $U_g = \{u_1, u_2, \dots, u_n\}$ è un insieme di utenti e $c_g \in I_g$ è il centro di g ;

- $\forall u \in U_g, u$ ha un certo interesse per I_g ;
- c_g è l'elemento con la minor distanza media dagli altri elementi di I_g e rappresenta l'interesse del gruppo;
- per ogni coppia ordinata di gruppi di interesse vale che $g_j (1 \leq i, j \leq k \wedge i \neq j); I_{g_i} \cap I_{g_j} = \emptyset \wedge U_{g_i} \neq U_{g_j}$; questa relazione significa che ad ogni coppia di gruppi vengono suggeriti oggetti differenti, e che, sebbene un utente possa far parte di più gruppi, a gruppi diversi non possono corrispondere gli stessi utenti di un altro gruppo.

Quanto descritto in queste righe è sufficiente per capire come vengono troncati i legami tra dati protetti e pubblici degli utenti. Non entreremo maggiormente nel dettaglio del funzionamento di Pistis, in quanto si basa su un meccanismo molto complicato e la spiegazione dei suoi dettagli esula dall'obiettivo che questa tesi si prefigge.

4.1.4 Risultati di Pistis

La valutazione di Pistis viene realizzata sulla base dei risultati ottenuti confrontandoli con altri tre algoritmi CF:

- *MinHash-based CF* (MCF), che prevede un'architettura scalabile e raggiunge un'accuratezza del suggerimento molto elevata;
- *SVD-Based* (SVD), un algoritmo recente, proposto da Canny [56], che adotta una crittografia omomorfa e decodifica multi-party per mantenere la riservatezza dei dati personali.
- *Basic-CF Method* (BCF), che utilizza esclusivamente i dati pubblici per la formulazione del suggerimento.

Non entreremo nel dettaglio del loro funzionamento, sebbene siano tre tra i più accreditati algoritmi di raccomandazione che riescono a mantenere nascosti i dati sensibili degli utenti.

Il confronto tra i quattro algoritmi viene fatto sulla base di tre componenti fondamentali:

- *qualità del suggerimento*: con Pistis si ottiene una notevole accuratezza rispetto agli altri 3;
- *riservatezza*: Pistis migliora la riservatezza del 157% rispetto a MCF e del 139% rispetto a SVD;
- *efficienza*: Pistis è distribuito, il che permette di avere un’architettura scalabile e un costo computazione ben diviso tra il client e il server. Il tempo di utilizzo di risorse del server, in media, è ridotto del 91% rispetto a MCF. Confrontato con SVD, che d’altro canto è un’applicazione P2P, Pistis riduce il costo computazionale del 26% e il carico a danno delle comunicazioni del 57% [55].

Osserviamo in figura 12 una sintesi ben chiara dei risultati ottenuti dagli esperimenti.

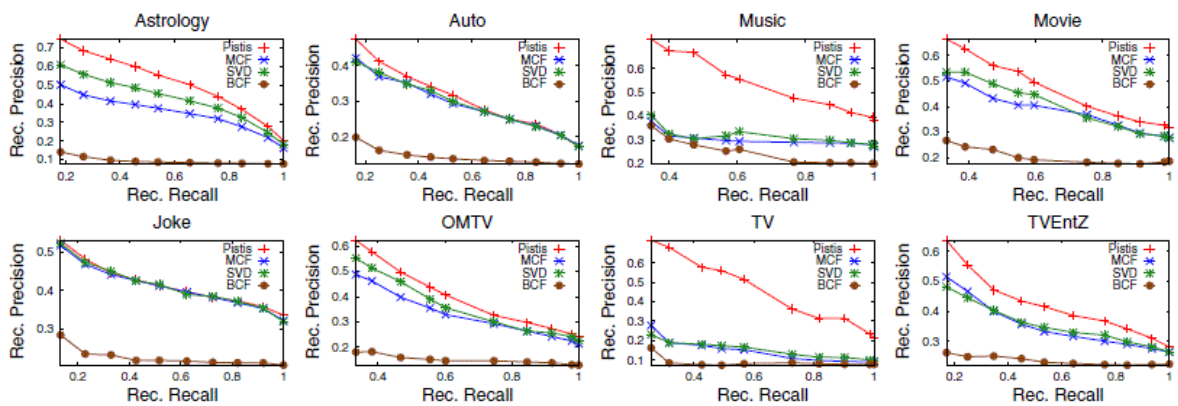


Figure 3. Recommendation quality of Pistis, MCF, SVD and BCF in eight subcommunities.

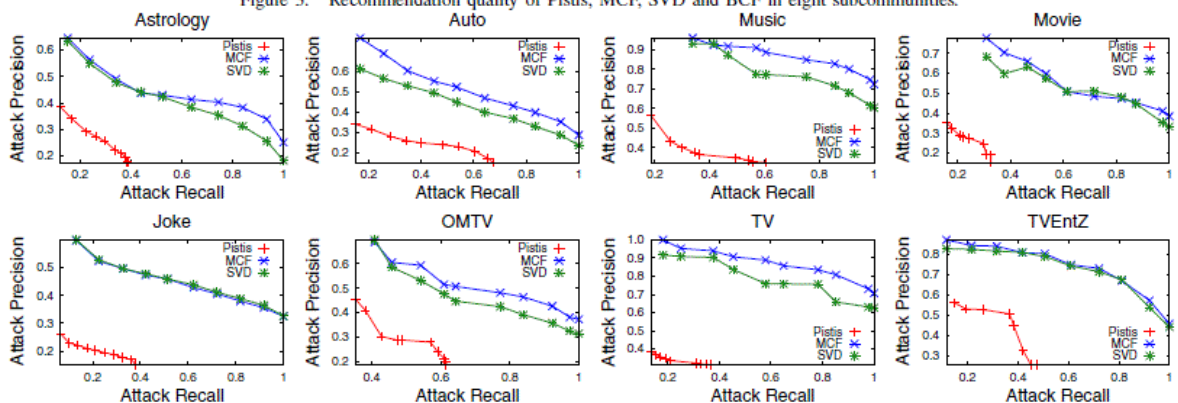


Figura 12 – Risultati di Pistis in confronto a MCF, SVD, BCF

Conclusione

Ormai i sistemi di raccomandazione sono entrati a far parte della vita di tutti i giorni. Li possiamo apprezzare ovunque, dal carrello della spesa di Amazon ® oppure nei banner pubblicitari a lato della Home Page di Facebook ®. Come abbiamo visto si adattano alla perfezione a qualunque ambito, arrivando persino a quello medico, suggerendo agli stessi dottori quali terapie siano da sottoporre al paziente in caso di precedenti ipersensibilità ai farmaci, togliendo quindi una grande mole di lavoro dalle loro spalle.

Abbiamo apprezzato la loro utilità nel suggerire al programmatore eventuali modifiche al software che sta analizzando, abbiamo analizzato in quale modo possono aiutare gli amministratori di rete nell'impostare firewall aziendali o domestici senza dover perdere una grande quantità di tempo per definirne le proprietà.

Ci siamo concentrati, quindi, sulle piattaforme di e-Learning, fondamentali nell'apprendimento a distanza. Sempre di più rapida diffusione, sono diventate il centro focale della diffusione della cultura su Internet, e rendono disponibili le nozioni anche a persone che per motivi di lavoro o economici non potrebbero studiare o prender parte a lezioni in classi reali.

Inoltre i sistemi di raccomandazione embedded nelle piattaforme e-Learning possono e dovrebbero essere sfruttati anche da comuni studenti, in modo da poter raggiungere più facilmente i loro obiettivi utilizzando metodi di insegnamento all'avanguardia ed adattivi per ciascuno di essi.

Il mondo frenetico in cui viviamo ci lascia poco tempo da dedicare alle cose che più amiamo. I sistemi di raccomandazione possono aumentare notevolmente la qualità del nostro tempo libero, è sufficiente conoscerli e imparare ad usarli.

Attualmente, sempre più studiosi si dedicano a questo campo, vista la facile portabilità del concetto di sistema di raccomandazione in ogni ambito. Lo scopo primario della ricerca in questo settore consiste nel trovare metodi di formulazione delle raccomandazioni a bassi costi computazionali che

analizzino quante più risorse disponibili in rete al fine di produrre un suggerimento migliore per l'utente.

In ambito e-Learning questo significa riuscire a combinare l'evoluzione tecnologica delle piattaforme, che nelle ultime versioni riesce quasi a riprodurre una classe reale, con metodi innovativi che permettono di veicolare lo studente in modo più preciso e affidabile.

È stato curioso analizzare come una tecnologia che richiede pochi interventi dell'uomo dopo essere stata messa in opera possa essere adattata alla medicina, settore in cui per decenni si è discusso sull'impiego di robot per eseguire operazioni chirurgiche che richiedono elevata precisione. Ebbene, come abbiamo visto, ADDRESS non elimina totalmente la figura del medico, bensì gli si affianca per alleggerire il suo carico di lavoro.

Questo esempio è di fondamentale importanza se si pensa a tutti i campi in cui l'uomo è ancora insostituibile, che se analizzati a fondo potrebbero portare all'introduzione della tecnologia come affiancamento per ottenere risultati in tempi ridotti e con calcoli più accurati.

Bibliografia

- [1] V. H. Resnick P., «Recommender Systems,» *Communications of the ACM*, pp. 56-58, 1997.
- [2] R. Burke, *Hybrid Recommender Systems: Survey and Experiments.*, 2002.
- [3] P. D. Bitonto, *Sistemi di raccomandazione E-learning*, 2011.
- [4] S. K. a. Y. K. Gediminas Adomavicius, «Improving Recommending Systems using filtering approach based on Rating Variance,» *IEEE*, pp. 734-749, 1995.
- [5] R. E., «User Modeling via Stereotypes,» *Cognitive Science* 3, pp. 329-354, 1979.
- [6] B. Krulwich, «LifeStyle Finder: Intelligent User Profiling Using Large-Scale,» *AI Magazine*, p. 37–45, 1997.
- [7] A. Moukas, «Amalthea: Information Filtering and Discovery Using a Multiagent Evolving System,» *Journal of Applied AI*, p. 437–457, 1997.
- [8] G. Boone, Interviewee, *Concept Features in RE:Agent, an Intelligent Email Agent*. [Intervista]. 1998.
- [9] M. &. B. D. Pazzani, «Learning and Revising User Profiles: The Identification of Interesting Web Sites,» *Machine Learners* , n. 27, p. 313–331, 1997.
- [10] M. Pazzani, «A Framework for Collaborative, Content-Based and Demographic Filtering,» *Artificial Intelligence Review*, n. 13, pp. 393 - 408, 1999.
- [11] V. P. T. R. V. R. P. Di Bitonto, «A taxonomy for cataloging LOs using IEEE educational metadata,» *Advanced Learning Technologies*, pp. 139-141,

2007.

- [12] K. Lang, «NewsWeeder: Learning to Filter News.».
- [13] N. J. a. C. W. B. Belkin, «Information Filtering and Information Retrieval: Two Sides of the Same Coin?,» *Communications of the ACM* , n. 35, pp. 29-38, 1992.
- [14] «CZ Live,» Luglio 2012. [Online]. Available: http://en.citizendium.org/wiki/Recommendation_system.
- [15] R. H. A. G. a. M. P. Guttman, «Agent-Mediated Electronic Commerce: A Survey,» *Knowledge Engineering Review*, vol. 2, n. 13, pp. 147-159, 1998.
- [16] J. C. Alan F. Smeaton, «Personalisation and recommender systems in digital libraries,» *Int J Digit Libr*, n. 5, p. 299–308, 2005.
- [17] A. & S. C. Stefani, *Personalizing Access to Web Wites: The Sitelf Project*, 1998.
- [18] B. & B. C. Krulwich, «AAAI Spring Symposium on Machine Learning in Information Access,» in *Learning User Information Interests Through Extraction of Semantically Significant Phrases.*, Stanford, CA., 1996.
- [19] B. a. C. P. Smyth, «A Personalized TV Listings Service for the Digital TV Age,» *Knowledge-Based Systems*, n. 13, pp. 53-59, 2000.
- [20] R. Burke, «Knowledge-based Recommender Systems,» *Encyclopedia of Library and Information Systems*, vol. 69, n. Supplemento 32, 2000.
- [21] C. H. H. a. C. W. Basu, «Recommendation as Classification: Using Social and Content-Based Information in Recommendation,» in *Proceedings in the 15th National Conference on Artificial Intelligence*, Madison, WI, 1998.
- [22] R. J. a. R. L. Mooney, «Content-Based Book Recommending Using Learning for Text Categorization,» in *Workshop on Recommender*

Systems: Algorithms and Evaluation, Berkley, CA, 1999.

- [23] G. F. Y. Ye, «Reuse-Conductive Development Environments,» *Automated Software Engineering*, vol. 12, n. 2, pp. 199-235, 2005.
- [24] R. W. G. M. R. Holmes, «Approximate Structural Context Matching: An Approach for Recommending Relevant Exaples,» *IEEE Trans. Software Engineering*, vol. 32, n. 1, pp. 952-970, 2006.
- [25] T. X. S. Thummalapenta, «PARSE Web: A Programming Assistant for Reusing Open Source Code on the Web,» in *IEEE/ACM Int'l Conf. Automated Software Eng.*, ACM Press, 2007, pp. 204-213.
- [26] M. Robillard, «Topology Analysis of Software Dependencies,» *ACM Trans Software Engineering and Methodology*, vol. 17, n. 4, 2008.
- [27] T. Z. e. al, «Mining Version Histories to Guide Software Changes,» *IEEE Trans Software Eng*, vol. 31, n. 6, pp. 429-445, 2005.
- [28] Y. W. H. S.-y. Y. S. K. T. L. Y. S. L. Yun-Jeong Bae, «The Effectiveness of Automatic Recommending System for Premedication in Reducing Recurrent Radiocontrast Media Hypersensitivity Reactions,» *PLOS ONE*, vol. 8, n. 6, 2013.
- [29] B. Mitchell, «Fire Wall - Network Firewalls,» 20 Maggio 2012. [Online]. Available: about.com.
- [30] J. Aaron, «Methods, Network Services and Computer Program Products for Recommending Security Policies to Firewalls,» *Computers, Network & Communications*, p. 505 in poi, 2012.
- [31] W. N. Y. Luming, *Modern Tourism E-Commerce Tutorial*, Pechino: Electronic Industry Press, 2004.
- [32] S. A. A. C. W. Lin, «Effective adaptive-support Association Rule Mining for Recommender Systems,» *Data Mining and Knowledge Discovery*, vol. 6,

pp. 83-105, 2002.

- [33] P. D. Bitonto, *Sistemi di Formazione a Distanza*, Bari, 2012.
- [34] C. J., «A model of school learning,» *Teachers College Record*, vol. 64, pp. 722-733, 1963.
- [35] P. V. Khuwaja R., «A model of tutoring based on the behaviour of effective human tutors,» in *Proceedings of the ITS*, Heildeberg, Germany, 1996.
- [36] A. C. A. L. B. Andronico, *E-Learning: metodi, strumenti, esperienze a confronto*, Napoli: Liguori Editore, 2002.
- [37] N. L. Reigeluth C.M., «A new paradigm of ISD?,» in *Educational media and technology yearbook*, Englewood, CO, Libraries Unlimited, 1997, pp. 24-55.
- [38] D. C. J. M. B. G. N. Porter, «Review and Recommendations for a Pan-Canadian Approach to Repository Implementation in Canada.,» in *A Report on Learning Object Repositories for CANARIE Inc*, 2002.
- [39] W. P.Y., «The Analysis and Design of Educational Recommender Systems,» in *Proceedings of Society for IT & Teacher Education International Conference*, Chesapeake, AACE, 2007, pp. 2134-2140.
- [40] C.-H. J. M. B. Castro-Herrera C., «A recommender systems for dynamically evolving online forums,» in *Proceedings of the Third ACM Conference on Recommending Systems*, New York, ACM, 2009, pp. 213-216.
- [41] M. G. Tang T, «Smart recommendation for an evolving e-learning system,» *International Journal of E-Learning*, vol. 4, n. 1, pp. 105-129, 2005.
- [42] K. G. K. V. Steinbach M, «A comparison of document clustering techniques,» in *KDD Workshop of Text Mining*, 2000, pp. 525-526.

- [43] S. B. R. X. Jaafar A, «Clustering analysis of railway driving mission with niching,» *The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 3, n. 31, pp. 920-931, 2012.
- [44] L. K. S. Richard M. Felder, «Learning and Teaching Styles In Engineering Education,» [Online]. Available: <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/LS-1988.pdf>.
- [45] J. C. G. ., *Tipi Psicologici*, Bollati Boringhieri, 1977.
- [46] Myer-Briggs, *Saggino Aristide La Teoria dei tipi psicologici*, La Terza Editore, 2002.
- [47] G. I. Carr B., «Overlays: A theory of modelling for computer aided instruction,» Cambridge, MA, 1977.
- [48] B. R. Brown J.S., «Diagnostic Models fore Procedural Bugs in Basic Mathematical Skills,» *Cognitive Science*, vol. 2, pp. 155-172, 1978.
- [49] N. D. Fleming, *Teaching and Learning Styles: VARK Strategies*, 2002.
- [50] O. S., «Constraint-based student modelling,» in *Student Modeling; the Key to Individualized Knowledge-based instruction*, 1994, pp. 167-189.
- [51] M. M. P. S. B. M. Antonija Mitrovic, «Constraint-Based Tutors: a Success Story Intelligent Computer Tutoring Group,» 2000. [Online]. Available: <http://www.cosc.canterbury.ac.nz/~tanja/cbmtut.pdf>.
- [52] M. L. P. Di Bitonto, «Un metodo di raccomandazione per ambienti e-Learning: la tecnica Rule-based,» *Journal of E-Learning and Knowledge Society*, vol. 6, n. 3, pp. 31-40, 2010.
- [53] R. A. M. N. J. Buncle, «A Recommendation Cascade for e-learning,» *IEEE*, pp. 740-747, 2013.

- [54] «Wikipedia,» 2013. [Online]. Available: <http://it.wikipedia.org/wiki/Privacy>. [Consultato il giorno 14 Novembre 2013].
- [55] Q. L. H. X. Dongsheng Li, «Pistis: A Privacy-Preserving Content Recommender Systems for Online Social Communities,» *IEEE*, pp. 79-86, 2011.
- [56] J. Canny, «Collaborative Filtering with privacy,» *S&P*, pp. 54-57, 2002.
- [57] W. H. Polat, «Privacy-preserving collaborative filtering using randomized perturbation techniques,» *ICDM*, pp. 625-628, 2009.
- [58] G. M. M. A. M. S. O. F. R. P. Capuano N, «an Intelligent Advisor for e-Learning Interactive Learning Environments,» *LIA*, vol. 17, n. 3, pp. 221-239, 2009.
- [59] R. I. M. G. N. Capuano, «A Recommender System for Learning Goals,» *WSKS*, pp. 515-521, 2013.

Ringraziamenti

Questa tesi rappresenta la fine del mio primo percorso universitario, ma anche l'inizio di un viaggio con molte aspettative.

Non ce l'avrei mai fatta senza l'aiuto di Mamma e Papà, che mi hanno sostenuto da sempre, in ogni mia incertezza e decisione, consigliandomi al meglio ma lasciandomi sempre libero di scegliere e di prendermi la responsabilità di quel che sceglievo. A loro non dedico questa tesi. Per loro è questa tesi.

Da poco più di un anno ho al mio fianco una persona stupenda, Alessia, che in poco tempo è diventata il centro della mia vita, per cui compio le mie azioni, da cui traggio ispirazione. Dice di non meritare questa dedica, ma non sono d'accordo. Senza di lei non sarei riuscito a dare 8 esami in tre mesi, a finire questo percorso. Mi è sempre stata vicino in questi mesi di stress e di studio "matto e disperatissimo" e le devo molto. Ti amo Boo.

La mia famiglia è molto ampia, parenti sparsi ovunque. Ma loro, i miei nonni, Nino e Lia, ci sono stati sempre, sebbene le differenze generazionali e il mio campo così tecnologico li rende un po' avulsi dai miei discorsi, provano sempre a capirmi. Vi voglio bene.

Da 11 anni viviamo sotto lo stesso tetto, combattendo il nostro comune nemico Mamma. Grazie di esserci sempre, di sopportarmi, di essere un buon orecchio e di sapermi consigliare sempre per il mio bene. Grazie Emilio.

A mio fratello Luca e alla sua dolce metà Ilenia, che lo tiene con i piedi per terra quando i geni dei Di Nardo prendono il sopravvento. A mio fratello che con i suoi modi "bruschi e animaleschi" dimostra di tenere a me e di volermi bene. Grazie

Ai miei amici più cari, Silvia e Poldo, Davide, Roberta e Federica, che hanno gioito con me, che hanno sofferto con me in questi tre lunghi anni di università. Siete unici, grazie.

Ogni professore ti lascia qualcosa, ma più di tutti l'uomo che mi ha segnato maggiormente è il Dott. Matteo Silimbani, tutor del corso di Logica Matematica, che in qualsiasi momento si è reso disponibile per aiutarmi e spiegarmi meglio dei concetti che esulavano anche dal suo corso di insegnamento. Indimenticabili le serate passate su Facebook mentre tentava di farmi capire le matrici di passaggio. Grazie

A Chiara, sua madre Franca e a Pupina che hanno pregato per me per permettermi di superare il mio ultimo esame in Novembre, consapevole del fatto che senza di loro non sarei mai riuscito in questa impresa titanica di verbalizzare Matematica Discreta.

A Uno, Mec, Sei, Losty che hanno reso le mie giornate cesenati decisamente piacevoli, che mi hanno aiutato nel preparare esami, che mi hanno fatto riappassionare a questi argomenti. Grazie amici miei.

A Claudia e al suo "ultimo giro di cooper" che mi ha fatto rimettere la testa sui libri a metà Settembre.

A tutti coloro che hanno creduto in me e che mi vogliono bene.