Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione
Corso di Laurea in Ingegneria dell'Informazione

Analisi e creazione di una collezione di test basata su giudizi graduati e commenti testuali per la valutazione di motori di ricerca musicali.

Candidato:
Michele Palmia

Relatore:
Prof. Giorgio Maria Di Nunzio

Anno accademico 2010/2011

Indice

Elenco delle figure

Sommario

La grande popolarità del Web e della rete Internet ha portato allo sviluppo di una grande quantità di informazioni prodotte nelle forme più diverse. Negli ultimi anni inoltre, la proporzione di informazioni generate dagli utenti stessi è aumentata considerevolmente con l'avvento del *Web 2.0*, ossia di applicazioni web che facilitano la condivisione di informazioni, la creazione di contenuti da parte dell'utenza e in generale il design incentrato sull'esperienza del singolo. La necessità di catalogare queste informazioni, ordinarle e renderle facilmente accessibili ha spinto fortemente ricerca e investimenti attorno all'ambito dell'*Information Retrieval*, che proprio di ciò si occupa.

Queste considerazioni generali valgono anche e ancora maggiormente per un ambito in grande mutamento come quello musicale: gli utenti, di fronte ad una moltitudine di possibilità e scelte, cercano, sempre più, sistemi e piattaforme che li aiutino a trovare ascolti graditi con facilità; artisti ed etichette discografiche sono interessate a far ascoltare la propria musica a chi sarà più propenso ad apprezzarla—e magari acquistarla.

Molte di queste piattaforme sono state studiate negli ultimi anni soprattutto al fine di sviluppare metodi migliori per fornire consigli musicali sempre più graditi agli appassionati. Nonostante ciò rimangono problemi insoluti e problematiche aperte, che lasciano spazio a nuovi approcci e metodologie. Solo ultimamente (a partire circa dalla seconda metà del primo decennio del 2000) informazioni non direttamente e strettamente legate alla musica e al feedback diretto degli utenti sono state prese in considerazione per migliorare le raccomandazioni musicali e i sistemi di ricerca. Proprio in questa direzione abbiamo deciso di spingerci, anche a causa della presenza di altri elaborati[1] che proponevano ricerche del genere.

Lo scopo della ricerca argomento dei prossimi capitoli è quello di analizzare una delle piattaforme discusse per estrapolarne dati utili ad analisi ed esperimenti in generale, sotto forma di un *test set*, un database, che sarà più ampiamente discusso più avanti. Dopo un'introduzione generale ai sistemi di *Information Retrieval* e *Music Recommendation* nel Capitolo 1, verranno descritti brevemente alcuni sistemi di *Information Retrieval* a sfondo musicale. L'analisi dei vari sistemi, la scelta di Amazon.com come la piattaforma,

[1] `http://tesi.cab.unipd.it/27643`

la progettazione concettuale e il passaggio alla progettazione logica del database sono analizzati in modo approfondito nel Capitolo 2. L'ultimo capitolo dell'elaborato verte sull'implementazione della base di dati, creata utilizzando dati recuperati da Amazon.com a partire da una lista di 500 album musicali. Il Capitolo 3 comprende anche una breve analisi dei dati ottenuti e alcune informazioni utili riguardo il loro utilizzo.

Capitolo 1

Introduzione

I concetti di *Information Retrieval* in generale e *Music Information Retrieval* in particolare sono di fondamentale importanza per proseguire con la trattazione degli argomenti specifici di questo elaborato; abbiamo perciò scelto di comporre il primo capitolo introduttivo di due sezioni corrispondenti ai due ambiti di ricerca appena citati: una prima sezione, dopo aver discusso la differenza tra *Data Retrieval* e *Information Retrieval* illustrerà i principali metodi usati per lo studio della rilevanza di documenti; la seconda affronterà lo studio dei sistemi di *Music Recommendation* e *Music Discovery*.

1.1 Information Retrieval

L'*Information retrieval* (IR) si occupa della rappresentazione, dell'immagazzinamento, dell'organizzazione e dell'accessibilità di contenuti informativi. Lo scopo di un sistema di *Information Retrieval* è quello di fornire agli utenti la possibilità di recuperare le informazioni desiderate nel modo più semplice ed efficace possibile.

Come evidenziato in Meadow *e altri* (1992) un utente che, ad esempio, osservi un canale televisivo di news per tutto il giorno ininterrottamente, senza esercitare controllo su ciò che sta venendo trasmesso non costituisce in alcun modo IR, mentre una biblioteca è l'esempio più semplice di sistema di questo tipo: un utente vi si recherà infatti con un idea particolare su ciò che desidera leggere e non per analizzare tutto il materiale in essa contenuto. Una biblioteca, come è evidente, fornisce un catalogo che permette un'organizzazione ottimale dei contenuti, permettendone l'accessibilità. Tale catalogo conterrà al suo interno una serie di informazioni che rappresentano il contenuto dei libri sotto forma di informazioni come ad esempio titolo, autore, anno di pubblicazione e casa editrice—alle quali comunemente ci si riferisce con il termine anglosassone *metadata*—unite ad una serie di parole chiave o di categorie.

Tale esempio evidenzia anche il fatto che un *Information Retrieval System* (IRS) non per forza consiste di un sistema informatico: le biblioteche hanno informatizzato i loro cataloghi solo da qualche decennio.

Finora abbiamo dato per scontato che il concetto di informazione fosse chiaro, ma basta pensare brevemente ai significati che questa parola può assumere per accorgersi che così non è. Spesso quando ci rivolgiamo ad un motore di ricerca abbiamo in mente quale tipo di informazione stiamo cercando—il risultato che vogliamo ottenere—ma non possiamo formulare una richiesta esattamente corrispondente a quella che abbiamo in mente, sotto forma di flusso di pensieri o di una domanda nel nostro linguaggio quotidiano: ce ne viene invece richiesta una traduzione sotto forma di *query* che il sistema possa interpretare. Comunemente, questa *query* consta di una serie di parole chiave che l'IRS interpreta per fornirci un insieme di documenti rilevanti al soddisfacimento della nostra ricerca, una richiesta diretta di dati più che di informazioni.

1.1.1 Data Retrieval o Information Retrieval

Vengono proposte qui di seguito due definizioni di dato e informazione, utili a capire meglio il significato reale di questi due termini nel contesto di sistemi di IR.

Dato Un dato (nella sua versione anglosassone[1] *datum*) è un insieme di simboli elementari. Essi non hanno per forza un significato evidente per ogni osservatore, ma deve essere chiaro a cosa facciano riferimento.

Informazione La definizione di informazione è più complessa perchè tale termine assume significati diversi nell'uso comune. Generalmente, informazione può fare riferimento a dati utili, universalmente accettati o validati da un esperimento, dati affidabili per qualche motivo. Shannon e Weaver (1959) definiscono come

$$i(A) = \log_2 \frac{1}{P[A]}$$

l'informazione associata ad un evento A: è questa la definizione formale di tale concetto. Basandosi sulla interpretazione intuitiva di informazione, è possibile ricavare tale definizione dal fatto che l'informazione (i) è sempre non negativa, (ii) è nulla per un evento certo, (iii) è minore per eventi più probabili e (iv) dati A e B due eventi, gode della proprietà $i(A + B) = i(A) + i(B)$. È possibile dimostrare che l'unica

[1] *Datum* è il singolare per *data* in Inglese. Tale lemma deriva dal participio passato del verbo latino *dare* e in tale linguaggio assume il significato di *dono*, *cosa data*.

legge che soddisfa tutti e quattro questi assiomi è una legge logaritmica come quella mostrata[2].

La definizio di informazione è naturalmente molto più estesa di questa ma il concetto dovrebbe essere sufficientemente chiaro per proseguire nell'indagine riguardo il rapporto tra *datum* e *information*. Il seguente esempio, oltre a chiarire ulteriormente la differenza tra i concetti, rende evidente il fatto che l'informazione riguardo un evento non è univocamente definita. Il *dato* riguardante la vittoria di una certa gara, infatti, può essere una *informazione* di valore molto alto per un appassionato qualunque, ma può avere scarso valore per chi ha truccato la partita in modo da essere sicuro della vittoria.

Nel contesto di sistemi di IR, recuperare dati significa, nel caso più semplice, determinare quali documenti in una collezione contengono le parole chiave presenti all'interno della *query* dell'utente. Nella maggior parte dei casi (Baeza-Yates e Ribeiro-Neto, 1999) ciò non è sufficiente a soddisfare l'utente, che ha bisogno di informazioni riguardo un argomento o una ricerca, più che di dati. Per essere efficace nel risolvere questo tipo di richieste, l'IRS ha la necessità di interpretare i documenti che rispondono meglio alla *query* per capire quanto essi siano effettivamente utili per l'utente. In questo modo il sistema avrà la possibilità di ordinare i risultati in modo che i primi, nella lista fornita all'utente, abbiano una rilevanza maggiore e siano, perciò, più informativi.

La nozione di rilevanza è fondamentale nel mondo dell'IR: alcuni dei più usati approcci per il calcolo della rilevanza e quindi del *ranking* (posizionamento) sono proposti nella prossima sezione.

1.1.2 Rilevanza dei documenti

Un sistema di *Information Retrieval* ha bisogno, come appena visto, di calcolare l'importanza dei documenti all'interno della propria collezione relativamente alle interrogazione dell'utente. Il risultato di questo calcolo può essere usato sia per decidere se il documento sia rilevante o meno—cioè se esso vada incluso nei risultati dell'interrogazione—sia per ordinare i risultati affinchè i più rilevanti e informativi siano evidenziati come tali. Il valore di questo calcolo è chiamato *retrieval status value* (RSV) ed è comunemente individuato da un numero reale che identifica l'affinità di un documento alla query. Tale valore può essere normalizzato, fornendo un numero reale compreso tra 0 e 1: il valore 0 corrisponde ad un documento senza alcuna relazione alla *query*, 1 è invece il valore adeguato ad un documento che risponde in modo perfetto ad essa. Naturalmente, esistono svariate modalità

[2]La definizione prevede che il logaritmo sia in base due cosicchè ad un evento con probabilità 1/2 è associata informazione unitaria, vedi Benvenuto e Zorzi (2010) per approfondimenti

per calcolare il RSV e questo è tutt'ora un'ambito di ricerca aperto. Qui di seguito vengono proposti solo tre dei più noti metodi.

Modello Booleano

Nel modello Booleano, sia la *query* sia i documenti nel sistema vengono interpretati come insiemi di termini. La *query* in questo caso consiste in un'espressione Booleana che indica la presenza o meno di termini nei documenti che si desidarano recuperare (ad esempio "musica $\wedge$ suono"—contestualizzata nel linguaggio opportuno—potrebbe essere sottoposta al sistema per ricevere documenti che contengano entrambe le parole); in questo caso il RSV può essere esclusivamente 1 (documento adatto) oppure 0 (documento non idoneo), poichè l'interrogazione richiede una risposta di tipo puramente booleano.

Modello Vettoriale

In questo tipo di modello, documenti e interrogazioni sono rappresentati come vettori numerici. Ogni elemento dei vettori corrisponde ad un determinato termine: se un termine appare nel documento, l'elemento ad esso relativo sarà non nullo; il valore dell'elemento viene invece calcolato tramite funzioni di pesatura che stabiliscono l'importanza di quel termine nel documento (*term weight*). Naturalmente ogni implementazione potrà differire (*i*) nella definizione di 'termine' come singola parola o piccolo gruppo di vocaboli, (*ii*) nella definizione del metodo per il calcolo del *term weight* e (*iii*) nella definizione di quali operazioni eseguire sui vettori per ricavare una misura per il RSV.

Una specifica largamente utilizzata per quanto riguarda il calcolo della distanza tra due vettori—in pratica una specifica per (*iii*)—è rappresentata dal *cosine similarity* (coseno di similitudine) e consiste nel calcolare il coseno dell'angolo formato dai vettori di documento e *query*:

$$\cos\theta = \frac{d \cdot q}{\|d\| \, \|q\|}$$

con d e q i vettori relativi al documento e alla query. Al numeratore è presente prodotto vettoriale tra i due, mentre al denominatore sono moltiplicate le rispettive norme. Questo valore è più semplice da calcolare rispetto a θ e fornisce un valore numerico facilmente interpretabile: nel caso $\cos\theta$ sia pari a 0, documento e query sono ortogonali e quindi non condividono alcun termine in comune; nel caso invece sia pari a 1, la corrispondenza è perfetta.

Nella versione originale del modello vettoriale (Salton e McGill, 1986) il peso dei singoli termini nei vettori dei documenti derivano sia da parametri locali che da parametri globali, secondo il modello chiamato *term frequency-inverse document frequency*. Il peso per un termine t nel documento d

secondo questo modello è calcolato come

$$w_{t,d} = \mathrm{tf}_{t,d} \cdot \log \frac{|D|}{|\{d' \in D \,|\, t \in d'\}|}$$

dove $\mathrm{tf}_{t,d}$ è la frequenza del termine t nel documento d, mentre la seconda parte del prodotto calcola la *inverse document frequency*, ossia il logaritmo dell'inverso del rapporto tra il numero di documenti che contengono il termine t e il numero totale di documenti.

Modello Probabilistico

Anche il modello probabilistico pesa i termini (il termine utilizzato è sempre *term weight*) per ricavare informazioni sulla rilevanza. In questo metodo viene assegnato un valore ad ogni documento per ogni termine della *query*, basato sulla probabilità che il documento sia rilevante relativamente a quel termine. Per aggregare i valori relativi a termini differenti, si ricorre normalmente all'uso del *log-odds*, un indice ricavato da quello appena descritto: detta p' la probabilità che un documento sia rilevate rispetto ad un primo termine e p'' la probabilità che lo stesso documento sia rilevante rispetto ad un secondo termine, tali due informazioni vengono associate utilizzando il logaritmo del prodotto di $p/(1-p)$ per ognuna delle probabilità, quindi

$$\log \frac{p'p''}{(1-p')(1-p'')}$$

che è possibile interpretare come somma di probabilità grazie alle proprietà del logaritmo.

1.2 Music Recommendation e Music Discovery

Finora si è sempre parlato in modo volutamente generico di documenti e di collezioni di dati. Ogni ambito specifico, all'interno della più ampia area di ricerca dell'*Information Retrieval*, ha caratteristiche differenti per determinare cosa sia un documento e ogni base di dati immagazzina documenti in forme diverse. Ogni sistema e applicazione tratta questi concetti in modo da ottimizzare le quattro proprietà fondamentali di un sistema IR viste nella sezione precedente, naturalmente rispettando le proprietà di ciò che si vuole rappresentare come documento.

Per chiarire ulteriormente il concetto, si pensi ad un sistema che immagazzina una grande quantità di brani musicali sotto forma di file mp3: è possibile stabilire che un 'documento', in questo caso, sia composto dal file stesso (o un riferimento al file), dal titolo del brano e dall'autore/esecutore, in modo che un utente del sistema possa cercare per autore o per titolo del brano e ricavarne il file. Visto che le tecnologie lo permettono, sarebbe possibile includere nella rappresentazione del documento anche il tempo

della traccia o altri parametri ricavati tramite un'analisi del file prima dell'inserimento nel sistema: ad un aumento di informazioni per l'utente, però, corrisponderebbe un maggior impiego di risorse.

L'ambito di ricerca della *Music Information Retrieval* (MIR) si occupa di applicare le tecniche note all'IR e di sviluppare nuove metodologie—nonchè di mutuarle da altri ambiti di ricerca—per la creazione di IRS in ambito musicale. Oggetto finale della ricerca degli utenti, in questo caso, sono le tracce audio: se in un sistema di *Information Retrieval* testuale il confronto tra *query* e documenti poteva essere fatto in modo diretto, ad esempio con i metodi visti prima, in un sistema nel quale i documenti sono semplicemente tracce audio, estrarre informazioni dalle stesse per soddisfare le ricerche degli utenti si fa sostanzialmente più complicato. Mentre per documenti di tipo testuale l'organizzazione da parte del sistema e la ricerca da parte degli utenti riguardava i termini presenti nel documento, per tracce audio il parallelo più immediato sarebbe quello con le singole note o con le melodie o con i fraseggi (ma non solo), tutte "entità" difficili da individuare con le attuali tecnologie. Per questo motivo—e l'argomento verrà ripreso anche in seguito—l'approccio più frequente ed immediato è risultato essere quello di raccogliere molte delle interazioni volontarie degli utenti con il sistema al fine di elaborarle e ricavarne consigli utili per gli altri utenti.

Tale ambito di ricerca ha subìto un grande sviluppo dopo che il *World Wide Web* ha permesso lo scambio di file e informazioni di tipo musicale in modo rapido ed efficiente, mentre la potenza dei calcolatori aumentava tanto da permettere le computazioni più complesse sulle tracce audio. Questa doppia spinta riflette i due principali approcci nella progettazione di sistemi di MIR: sistemi basati sul contesto contro sistemi basati sul contenuto, che verranno analizzati brevemente nel paragrafo successivo. D'altra parte, anche gli utenti hanno visto accrescere le proprie possibilità di scoprire nuova musica, in una varietà di nuovi approcci ai problemi di *Music Discovery* e *Music Recommendation*, molto studiati negli ultimi anni sia dalla comunità scientifica che da un folto gruppo di aziende. Queste ultime forniscono servizi (alcuni dei quali sono analizzati nei capitoli successivi) che commercializzano sia verso gli utenti, che trovano beneficio nello scoprire nuovi artisti e nell'ascoltare nuovi brani di loro gradimento, sia alle etichette discografiche e ai musicisti, che possono proporre la propria arte ad ascoltatori selezionati.

1.2.1 Metodi per Music Recommendation

Il problema della creazione di suggerimenti musicali personalizzati, analizzato da Celma (2008) in grande dettaglio, può essere intuitivamente diviso in due sottoproblemi. Il primo è un problema di predizione, e riguarda la stima di quanto un prodotto (o, nell'accezione usata prima, un documento), sia adeguato ad un certo utente. Il secondo problema è quello di fornire una lista di prodotti—analogamente a quanto avveniva per i sistemi di IR.

Esistono diversi approcci alla soluzione di questo problema, alcuni dei quali vengono proposti di seguito. Tutti questi sistemi sfruttano le proprie conoscenze relative all'intera comunità di utenti per fornire consigli mirati e massimizzare l'esperienza del singolo, che a sua voltà fornira *feedback* che sarà poi sfruttato per fornire consigli utili all'intera comunità. Qui presentiamo 4 dei metodi più utilizzati per fornire consigli musicali mirati agli utenti; tutti i 4 sistemi illustrati sono basati su funzioni, comunemente chiamate "filtri", che selezionano con differenti metodi un numero limitato di elementi, da fornire in risposta ad una *query* o come generiche raccomandazioni, a partire dal loro intero catalogo.

Filtro demografico

Un filtro di questo tipo può essere utilizzato per identificare l'"utente tipo" che gradisce un certo prodotto (ad esempio lo stereotipo di utente al quale piace un certo gruppo musicale). Questa tecnica classifica gli utenti in gruppi in base a dati personali (come età, genere etc.), dati geografici e dati psicografici (interessi, stile di vita, generalmente forniti dall'utente stesso).

Filtro collaborativo

L'approccio tramite *collaborative filtering* (o CF per brevità) consiste nello studiare interazioni passate tra utenti e prodotti al fine di proporre consigli studiati a utenti che hanno interazioni simili o compatibili. Il CF è un ambito di ricerca tanto esteso da formare un'area completamente a se stante, si rimanda a testi specifici (Breese *e altri* (1998) e Sarwar *e altri* (2001) illustrano molti algoritmi di questo tipo) per l'approfondimento.

Sistemi basati sul contenuto

I sistemi basati sul contenuto (nella letteratura anglosassone *content-based*) si basano sull'analisi dei prodotti—ad esempio sull'analisi algoritmica di file musicali—con lo scopo di estrarne informazioni utili (per le tracce audio, ad esempio, tempo, volume[3], tonalità etc.). I sistemi *content-based* sono utili per la ricerca di brani con melodie o fraseggi specifici, brani particolarmente "rumorosi" o lenti e così via. Tali sistemi sono anche necessari quando le annotazioni ai prodotti—nome del brano, dell'artista, anno di pubblicazione etc.—sono completamente assenti mentre possono sensibilmente migliorare i risultati delle ricerche quando tali annotazioni sono presenti.

Sistemi basati sul contesto

Il contesto è qualsiasi informazione che possa essere utilizzata per caratterizzare la situazione di un entità (Abowd *e altri*, 1999). Sistemi basati sul

[3] *Loudness* è il termine comunemente utilizzato per indicare il livello di volume.

contesto, quindi, utilizzano informazioni contestuali per descrivere il prodotto. Per quanto riguarda tracce audio, ad esempio, è possibile analizzare documenti e ipertesti in qualche modo correlati a tale traccia per ottenere da essi (ad esempio con metodi e modelli di IR visti sopra) parole chiave per la traccia. Alcuni esempi di filtri *context-based* sono il **web data mining**, ossia l'analisi delle informazioni sulla navigazione dell'utente, delle sue abitudini ed usanze, del suo rapporto con prodotti specifici e delle sue preferenze e il **social tagging**, che utilizza l'annotazione collaborativa dei prodotti da parte degli utenti, che annotano (associano *tag*, cioè una singola parola o una breve frase) ai prodotti in modo libero.

Capitolo 2

Analisi dei requisiti

Nella prima parte di questo capitolo verranno presentate alcune popolari piattaforme di *Music Discovery*—questo termine, come si vedrà in seguito, è qui usato in senso lato—subito dopo una breve introduzione al problema da risolvere; in una seconda parte si procederà alla descrizione di una di queste piattaforme in particolare, non prima di aver illustrato i motivi di tale scelta. La terza parte riguarda la progettazione concettuale del database le cui caratteristiche verranno illustrate nei prossimi paragrafi mentre la quarta verte sulla traduzione di tale modello concettuale in modello logico relazionale.

Nel capitolo precedente è stato evidenziato come lo sviluppo e la progettazione di sistemi di *Information Retrieval* abbia ricevuto una notevole spinta produttiva a partire dalla fine del secolo scorso, grazie all'avanzamento delle tecnologie per la trasmissione dati in generale e in particolare grazie alla diffusione di internet. Ampia attenzione è stata dedicata negli ultimi anni all'ambito della *Music Information Retrieval* e più in particolare della *Music Discovery*: come dato del tutto qualitativo, si noti che il numero di articoli segnalati da Google Scholar[1] contenenti le parole "music information retrieval" o "music recommendation" all'interno del titolo è di 51 per il solo 2010, mentre per l'intero decennio 1990–2000 arriva solamente a 10. Tale inedito interesse è cresciuto soprattutto grazie alla sopraggiunta possibilità di trasmettere in streaming brani e video musicali senza lunghe attese e alla possibilità di navigare in internet anche attraverso dispositivi mobili.

Parallelamente allo studio di questi sistemi da parte del mondo accademico, sono state sviluppate una serie di piattaforme per l'esplorazione musicale gestite da aziende private che hanno negli ultimi tempi attirato un grande pubblico e notevoli capitali. La maggioranza delle più note ed utilizzate piattaforme forniscono a sviluppatori e ricercatori API[2] attraverso le quali integrare servizi o recuperare dati. Oltre a piattaforme di questo

[1] `http://scholar.google.com`

[2] *Application Programming Interface*, una serie di regole attraverso le quali è possibile comunicare con l'applicazione.

tipo, progettate dalla nascita come *Music Recommender Systems* (sistemi di raccomandazione musicale, che forniscono cioè consigli musicali), anche piattaforme di vendita elettronica come Amazon.com[3] o Play.com[4] gestiscono dati simili o paragonabili[5] a quelli di un sistema di scoperta musicale e verranno perciò proposti nella seguente analisi.

Come spiegato nell'introduzione, spesso i *MR Systems* si basano su filtri di vario tipo, utilizzando la grande disponibilità di dati a loro disposizione, relativi a utenti differenti, per valutare consigli più appropriati per gli utenti stessi: molti di questi dati sono resi disponibili a ricercatori e programmatori tramite numerose, gia' citate, API . Proprio grazie a ciò, alcune di queste piattaforme sono state largamente studiate nel mondo accademico: Turnbull *e altri* (2008), Barrington *e altri* (2009) e Miotto e Orio (2010) sono solo alcuni esempi di studi che utilizzano e confrontano anche questo tipo di informazioni. I dati forniti da alcune di queste piattaforme vengono talvolta aggregati a formare un'unico database pronto a qualsiasi tipo di utilizzo se ne voglia fare: esattamente ciò che il progetto oggetto di questo elaborato si pone come obiettivo. Un esempio di questo tipo che merita certamente una menzione è il Million Song Dataset[6], un database di dati raccolti riguardo un milione di tracce audio utilizzando le API di EchoNest, servizio discusso in uno dei prossimi paragrafi.

Uno dei problemi avvertiti lavorando con dati provenienti da queste piattaforme è che essi constano per la maggior parte di giudizi di tipo positivo: i dati di ascolto riguardano le tracce effettivamente ascoltate, i commenti riguardano i musicisti che amiamo o che ci interessano per qualche motivo, dei quali ci siamo goduti un bel concerto, e che vogliamo magari ringraziare attraverso la piattaforma. Naturalmente ciò non è vero in maniera assoluta ed è possibile trovare commenti negativi nei luoghi e nelle forme più svariati, ma trovare una soluzione a questo "problema" potrebbe essere utile in diversi modi: proprio alla ricerca di materiale di tipo differente da quello già analizzato da precedenti lavori si prosegue con l'elaborato.

2.1 Siti web per l'esplorazione musicale

In questa prima sezione presenteremo alcuni dei più noti servizi e piattaforme per l'esplorazione musicale. Oltre alla descrizione del servizio offerto, nei se-

[3] `http://www.amazon.com`

[4] `http://www.play.com`

[5] Sia nei casi di *Music Recommendation* più propriamente detto che in casi di *e-commerce* musicale, gli utenti sono portati a scoprire nuova musica sulla base di ascolti precedenti o appartenenza a gruppi demografici specifici: alcune tecniche sono già state analizzate nel capitolo precedente. La prima tipologia di piattaforme ha questo compito (fornire consigli mirati agli ascoltatori) come prerogativa, essendo nata per soddisfare questo "bisogno", mentre la seconda tipologia può ottenere maggiori guadagni se fornisce agli acquirenti, ascoltatori, dischi appetibili.

[6] `http://labrosa.ee.columbia.edu/millionsong`

guenti paragrafi cercheremo di evidenziare le peculiarità di ogni piattaforma e il possibile uso dei dati recuperabili.

Last.fm[7] è un servizio di radio personalizzate online. La piattaforma è nata nel 2005 dalla fusione di un servizio di radio personalizzate—Last.fm (ideato nel 2002)—e uno di registrazione degli ascolti—conosciuto come AudioScrobbler (creato nel 2003). Last.fm registra ogni singolo ascolto dell'utente (tramite plugin per applicazioni di terze parti o integrazioni su altre piattaforme) in un processo chiamato *scrobbling*: l'elevata quantità di dati così a disposizione tramite questo processo di raccolta dati viene utilizzata sia per stabilire in modo empirico la somiglianza tra artisti, brani ed album all'interno del sistema sia per fornire all'utente consigli mirati sugli ascolti.

Gli artisti, le etichette e gli stessi utenti sono invitati ad arricchire le pagine dedicate alle band ed ai musicisti con materiale fotografico e biografico. Gli utenti sono anche invitati ad associare *tag* che esprimano con pochi caratteri una caratteristica dell'artista, del brano o dell'album in questione; non essendo presente nessuna limitazione nella scelta di tali *tag*, essi presentano una grande variabilità e sono purtroppo soggetti a diversi problemi (Lamere e Celma, 2007), anche se restano comunque una interessantissima fonte di informazioni. Oltre le varie radio personalizzate, la principale attrattiva della piattaforma è il sistema di "consigli" che propone all'utente musica da ascoltare in base ai suoi gusti.

Le radio di Last.fm sono attualmente disponibili gratuitamente solo negli Stati Uniti d'America, Regno Unito e Germania, unitamente ad un servizio a pagamento di migliore qualità. Il servizio di scrobbling e di raccomandazione di nuovi artisti è invece disponibile in tutto il mondo. La piattaforma fornisce comunque in modo del tutto gratuito l'accesso a una grande quantità di dati tramite API[8] di semplice utilizzo. Attraverso tali interfacce è possibile accedere, ad esempio, alla lista dei commenti o dei tag relativi ad album, artisti o singole canzoni, a svariate classifiche di ascolto relative a singoli periodi di tempo e molto altro. Last.fm è probabilmente il sistema di *music recommendation* più citato e utilizzato dal mondo accademico.

Pandora[9] è un servizio di radio personalizzate online, attualmente disponibile solo negli Stati Uniti d'America, basate sui dati forniti da esperti musicali. Ogni brano proposto all'utente è infatti contrassegnato da tag musicalmente oggettivi (secondo i creatori del sistema) assegnati appunto da esperti: il sistema richiede in input il nome di una traccia o di un artista (*seed*) e genera di conseguenza, utilizzando tali conoscenze, una lista di brani che l'utente può ulteriormente modellare a suo piacere aggiungendo

[7] `http://www.last.fm`
[8] `http://www.lastfm.it/api`
[9] `http://www.pandora.com`

variability alla stazione appena generata (o salvata da una precedente "sessione" di ascolto) oppure valutando positivamente una canzone particolarmente azzeccata o negativamente una indesiderata.

Le notizie riguardanti le annotazioni dei brani sono state recuperate dalle FAQ[10] presenti sulla piattaforma, che come ci si potrebbe aspettare non sono particolarmente generose nei dettagli tecnici. Alcune annotazioni sono comunque disponibili sulle pagine dedicate agli artisti, facilmente recuperabili tramite un *web crawler* ad hoc. In Tingle *e altri* (2010) si fa notare come queste annotazioni siano normalmente più neutre dei tag associati ai brani dagli utenti come ad esempio quelli ricavabili da Last.fm; ciò rende particolarmente appetibili queste annotazioni come "informazioni di controllo" visto che in teoria esse identificano un giudizio musicale oggettivo.

Soundcloud[11] è una piattaforma per la condivisione di file musicali. Definito il YouTube per la musica, sta diventando una primaria via di comunicazione da parte degli artisti per promuovere le proprie produzioni e da parte degli ascoltatori per ascoltare e scaricare brani inediti legalmente.

Le tracce sono annotate dall'autore con un genere musicale specifico e una serie di *tag*, mentre gli utenti possono commentare la traccia con commenti generici o inseriti in uno specifico punto del brano. Nonostante tali informazioni siano liberamente disponibili tramite una API gratuita[12], sembra che solo di recente la ricerca stia cominciando ad interessarsi a questa piattaforma (si rimanda ad esempio a Bogdanov *e altri* (2011), che comunque usa il sistema con un'ottica *content-based*). Essendo possibile scaricare fisicamente le tracce per una grande quantità di utenti collegati tra loro attraverso gruppi, amicizie e commenti potrebbe rivelarsi interessante analizzare come e se le interazioni fisiche all'interno del sito web corrispondono a somiglianza musicale, ad esempio.

Youtube[13] consente la condivisione di filmati tra gli utenti. E' stato stimato che circa il 31% dei video presenti sulla piattaforma siano effettivamente video musicali[14], nonostante la maggioranza non sia pubblicata in modo ufficiale dalle etichette o dagli artisti stessi, bensì da fan e appassionati.

Anche su questa piattaforma, analogamente a quanto avviene su SoundCloud, il materiale caricato è annotato dall'utente che procede all'upload, mentre agli utenti in generale è data la possiblità di lasciare commenti sia sotto forma di breve testo che di voto numerico (attualmente binario). Anche ogni singolo commento può essere votato in maniera numerica binaria.

[10] `http://blog.pandora.com/faq/`
[11] `http://www.soundcloud.com`
[12] `http://developers.soundcloud.com/`
[13] `http://www.youtube.com`
[14] `http://virtualmusic.tv/2011/02/2010-music-website-heat-map/`

Youtube fornisce consigli relativi a video che potrebbero essere di gradimento all'utente, anche musicalmente, sulla base di video guardati di recente. Nonostante vengano fornite API[15] anche per questa piattaforma, questo tipo di dati sulla relazione tra singoli video non vengono resi disponibili. È invece possibile recuperare tutti i commenti relativi ad ogni video, ma purtroppo questi ultimi sono spesso poco in relazione con il video stesso e proprio a causa di ciò di scarso interesse per quanto riguarda la ricerca.

Wikipedia[16] nota enciclopedia libera—disponibile in 270 lingue—ospita tra i suoi numerosissimi articoli una grande quantità di pagine riguardo artisti, album, dischi, financo singoli brani. La sua principale prerogativa, quella essere un documento collaborativo, è anche la principale caratteristica che interessa l'*Information Retrieval*: gli articoli, nella filosofia dell'enciclopedia, sono redatti in modo da descrivere personaggi e opere letterarie, musicali o di qualsiasi altro tipo senza cercare di stabilire verità oggettive.

È lampante come un'enciclopedia non si possa definire in senso stretto un sistema di *Music Recommendation*, ma basta inoltrarsi poche righe nella lettura di una voce con argomento musicale per accorgersi che la fitta rete di collegamenti ipertestuali rende facile passare da una pagina all'altra e scoprire nuovi artisti, brani, album. Anche Wikipedia fornisce un'interfaccia per il download dei documenti[17] e applicando metodi e modelli per lo studio dei documenti, alcuni dei quali già visti nello scorso paragrafo, a voci di interesse, è possibile estrarre interessanti informazioni sulla correlazione tra le pagine dell'enciclopedia e quindi tra artisti, tra singoli brani e così via.

Amazon.com[18] è una societa' di commercio elettronico: fondata nel 1995 come libreria[19] on line, ha presto ampliato l'offerta dei prodotti venduti includendo CD musicali, DVD, prodotti elettronici, abbigliamento e molto altro. La presenza di materiale musicale sul sito è molto forte: è possibile trovare i cataloghi integrali della larga maggioranza degli artisti internazionali, perfettamente catalogati e organizzati.

Ogni prodotto ha una pagina specificamente dedicata, che include tutte le informazioni utili all'aquirente, alcune delle quali verranno analizzate in maniera più approfondita in seguito, e una serie di commenti di altri acquirenti riguardo il prodotto. Questi commenti, diversamente da quelli lasciati sulle altre piattaforme già citate, sono usualmente di alto livello, potendo essere lasciati solo da acquirenti del prodotto stesso. Inoltre, ad ogni commento sono associate svariate informazioni di contesto.

[15] `http://code.google.com/apis/youtube`
[16] `http://www.wikipedia.org`
[17] `http://www.mediawiki.org/wiki/API`
[18] `http://www.amazon.com`
[19] Nel significato di *negozio di libri*.

Ciò che rende ancora più appetibili tali commenti è la presenza di più livelli di "valutazione" del prodotto all'interno di uno stesso commento. I due livelli più evidenti—alcuni altri saranno citati o approfonditi in seguito—sono quelli forniti dal commento testuale abbinato al commento numerico: associata ad ogni recensione di un prodotto, infatti, oltre alla classica recensione viene anche richiesta una valutazione numerica (da 0 a 5) che dovrebbe appunto corrispondere alla valutazione poi compiutamente espressa nel commento stesso che compone la recensione del prodotto.

Se siamo solitamente portati a lasciare commenti positivi nel caso la possibilità di dire la nostra sia del tutto libera, lasciare un commento negativo in cui spieghiamo perchè pensiamo di aver speso male i nostri soldi è probabilmente più naturale. Se ci si pensa, in effetti, l'insieme di ciò che ci piace e del quale siamo a conoscenza è certamente molto più ampio dell'insieme di ciò che non ci piace: nella maggior parte dei casi, non conosciamo nemmeno i nomi degli artisti che non ci piacciono, semplicemente perchè nessuno ce li ha mai consigliati, non ne abbiamo mai sentito parlare e non abbiamo a disposizione il tempo necessario a provare ascolti di tutti i generi. Quando interroghiamo sistemi di *Music Information Retrieval*, in effetti, stiamo probabilmente cercando artisti che ci aggradino: il sistema deve evitare tutto ciò che corrisponde a caratteristiche contrarie a quelle che stiamo proponendo come termine di paragone. Ma se il ragionamento per giungere a questo punto è complicato per un umano, si pensi a quanto può essere inesatta la soluzione fornita da un computer.

Avendo a disposizione commenti dichiaratamente negativi, allora, potrebbe essere possibile valutare se un IRS di tipo musicale "stimi" in modo adeguato le sue risposte (o, sarebbe meglio dire, le sue *non risposte*). Essendo questo intero compito necessariamente troppo azzardato per un elaborato di questo livello, ci si è limitati alla creazione di un limitato insieme di commenti provenienti da Amazon.com utile a una futura valutazione; per questo tale database viene normalmente chiamato *test set* o *test collection*.

Aggregatori e altre API Dai primi anni 2000 la rete ha assistito ad una crescita notevole nel numero di blog di ambito musicale, moltissimi dei quali forniscono gratuitamente e a solo scopo promozionale la possibilità di ascoltare e scaricare novità discografiche. Per far fronte al crescente numero di tali weblog sono stati sviluppati alcuni *aggregatori*, applicazioni web utili a raggruppare su un'unica pagina o comunque in modo efficiente la grande mole di informazioni prodotta in ogni istanti dai redattori di blog. *The Hype Machine*[20] e *We Are Hunted*[21] sono due differenti esempi di questo approccio che, seppur in maniera completamente differente, forniscono agli utenti contenuti aggregati provenienti dai più disparati blog.

[20] `http://www.hypem.com`
[21] `http://www.wearehunted.com`

Alcuni di questi aggregatori—WAH in primis[22]—forniscono API attraverso le quali è possibile scaricare le informazioni elaborate dall'aggregatore, ad esempio sotto forma di classifiche giorno per giorno o settimana per settimana, anche per singolo genere, delle tracce o degli artisti dei quali più si è parlato nei blog di tutto il mondo. Purtroppo questi dati sono spesso limitati ad una nicchia musicale che difficilmente si sovrappone bene al mercato musicale generalista.

Da notare anche i servizi che EchoNest[23], GraceNote[24], MusicBrainz[25], DiscoGS[26] e altri forniscono in maniera—chi più chi meno—gratuita. EchoNest e GraceNote sono applicazioni web pensate per sviluppatori che non sono dotate di interfaccia utente ma solo di Application Programming Interface: permettono cioè l'integrazione di funzioni specifiche, che in questo caso sono la ricerca di dati sugli artisti e la somiglianza tra di essi, l'identificazione di tracce audio e molto altro, attraverso linguaggio facile da interpretare con i più comuni linguaggi di programmazione e proprio per questo facile da integrare nelle più disparata applicazioni. MusicBrainz e DiscoGS invece (ma sono solo due esempi) sono cataloghi musicali (piuttosto dettagliati) gestiti dagli utenti e forniti di API (ancora una volta più o meno) gratuite. Da tutte queste interfacce potrebbe essere possibile ricavare altri tipi di dati che non è opportuno approfondire qui. EchoNest ha anche recentemente reso disponibile un sistema chiamato EchoPrint[27], che permette agli sviluppatori di integrare in maniera del tutto gratuita il riconoscimento di tracce audio all'interno di applicazioni di qualsiasi tipo: sistemi come questo stanno progressivamente facendo spazio ad applicazioni *content-based* più che *context-based*.

2.2 La struttura di Amazon

Nei paragrafi della sezione precedente all'interno dei quali abbiamo approfondito la natura di Amazon è stato evidenziato come al termine dell'analisi dei vari servizi illustrati sia stata proprio quest'ultima la piattaforma scelta per la creazione di un database, il già citato *test set*. Nel corso di questa sezione è illustrata in dettaglio la struttura di Amazon.com.

Ogni prodotto all'interno di Amazon.com è associato ad una particolare pagina—naturalmente ciò vale anche per gli album musicali e per ogni singola traccia in formato mp3—all'interno della quale sono contenute molte delle informazioni utili in forma breve o abbreviata, unitamente ad un link per l'approfondimento di tali contenuti: sono presenti ad esempio solo i commenti

[22] `http://wearehunted.com/a/#/api`
[23] `http://developer.echonest.com`
[24] `http://www.gracenote.com`
[25] `http://musicbrainz.org`
[26] `http://www.discogs.com`
[27] `http://echoprint.me`

considerati più utili e l'incipit di quelli più recenti, accanto ai riferimenti per leggere l'intera lista di commenti relativi al prodotto. Ogni "tipologia" di pagina è di tipo standard, identica ad ogni altra dello stesso tipo eccetto per i contenuti. Ogni categoria di prodotti ha le proprie caratteristiche specifiche e proprie "pagine di sevizio" adatte alle esigenze specifiche della categoria (tipologie differenti di pagine). Nei prossimi paragrafi verrà analizzata solo l'organizzazione dei prodotti di tipo *Audio CD*.

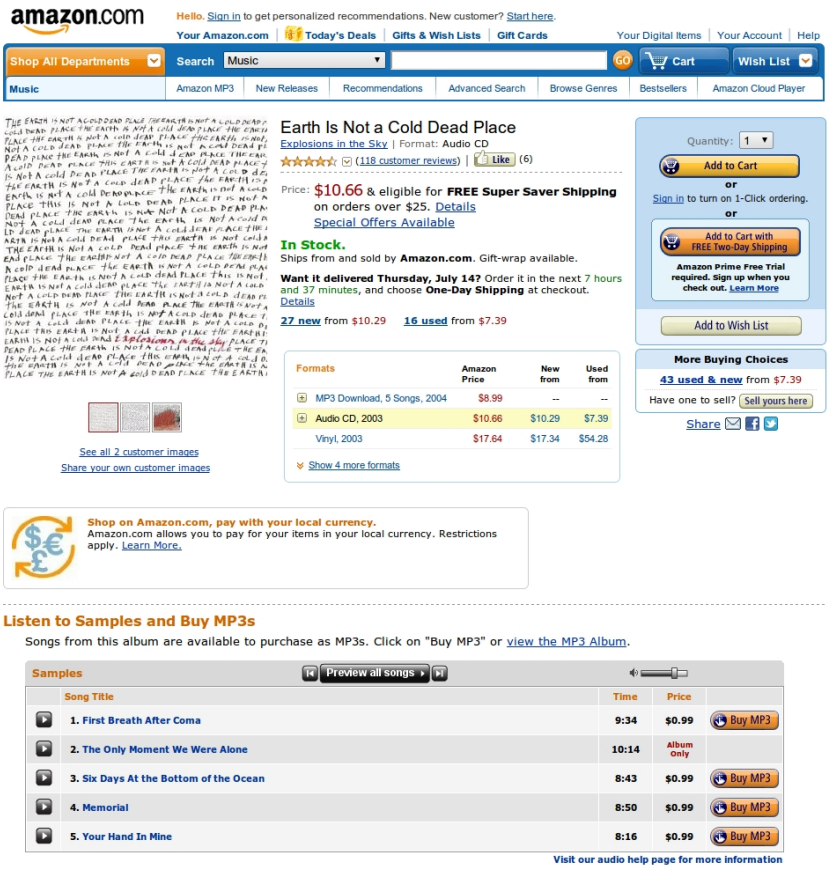

Figura 2.1: *Screenshot* della prima parte della pagina di un prodotto di tipo *Audio CD* su Amazon.com. Nella parte bassa è presente anche la parte dedicata al *Track Listings*, discussa nella sezione 2.2.2

Le entità centrali, di partenza, utilizzate nella creazione del *test set* infatti sono state le pagine relative agli album: non appena individuato l'obiettivo da raggiungere, già presentato nell'ultimo dei paragrafi riguardo Amazon nella sezione 2.1, si è reso evidente il fatto che le pagine relative ai singoli brani non contenevano sufficienti commenti. La possibilità di usare commenti di questo tipo avrebbe fornito più gradi di libertà nelle successive analisi,

perchè la singola traccia è l'unità base dalla quale è più semplice ricostruire valutazioni per album e artisti, ma non è appunto stato possibile continuare con questo approccio per i motivi appena citati.

Nel seguito saranno spesso presenti cenni riguardo la formattazione del codice delle pagine presentate perchè, come spiegato nel capitolo successivo, per recuperare le informazioni necessarie alla compilazione del database è stato usato un parser sulle singole pagine HTML. In ogni sezione sarà anche presentata la struttura dell'URL di riferimento per la risorsa discussa. La pagina dalla quale si dirama tale analisi è quella relativa al prodotto, la pagina relativa ad ogni singolo *Album*; quasi tutte le pagine di questo tipo contengono un riferimento alla pagina dei *Track Listings*, dov'è contenuta la lista delle tracce contenute nell'album (o negli album se il prodotto consta di più di un disco), la seconda tipologia di pagine illustrata in questa sezione. Successivamente verrà illustrata la formattazione standard per i *Commenti* relativi ad ogni album e per la pagina relativa al *Profilo Utente*, raggiungibile per ogni commentatore.

2.2.1 Album

Per quanto riguarda la sezione musicale, il nome del prodotto corrisponde nella maggior parte dei casi al nome dell'album come scelto dalla band. Spesso insieme al nome sono presenti l'anno di riferimento del disco e una serie di informazioni supplementari sul prodotto fisico (è un disco originale o una rimasterizzazione, contiene tracce supplementari o inedite e così via). È comune trovare esempi di questo tipo in qualsiasi album acquistabile, soprattutto se datato. Se si visualizza ad esempio la pagina relativa all'album *Songs From the Wood*, della band inglese *Jethro Tull*[28] si nota immediatamente che il nome del prodotto è presentato come Songs From the Wood [Original Recording Reissued, Original Recording Remastered, Extra Tracks]. Fortunatamente, è piuttosto semplice sia per un umano che per un software capire quale è effettivamente il titolo e quali sono le proprietà del prodotto, essendo queste separate del titolo stesso nel codice della pagina e presentate in formato leggermente più piccolo.

Sotto e intorno al titolo sono presenti altre informazioni utili, come l'autore cui fa riferimento il prodotto, il prezzo e i consigli per gli aquisti simili. Proseguendo invece verso la parte centrale della pagina è presente la lista delle tracce presenti nel disco venduto e una serie di informazioni specifiche sul prodotto (codice di catalogo, anno di pubblicazione, categorie di vendita etc.). Ancora più in basso sono presenti alcuni commenti unitamente ai collegamenti necessari a raggiungere la pagina ad essi specificamente dedicata.

[28] `http://www.amazon.com/Songs-Wood-Jethro-Tull/dp/B00008G9JN`

URL L'identificatore univoco dei prodotti su Amazon (almeno per quanto riguarda la sezione musicale) è una sola stringa alfanumerica, mentre esistono almeno due differenti versioni di URL per ogni singolo prodotto[29]. I link forniti dal motore di ricerca interno di Amazon sono composti da una stringa di caratteri e trattini (-) associata al prodotto (nel caso dell'esempio precedente Songs-Wood-Jethro-Tull), che in realtà identifica un insieme più o meno grande di album che differiscono solamente nell'edizione, e dall'identificativo alfanumerico illustrato sopra. Del già citato album *Songs From the Wood*, ad esempio, sono presenti due versioni: solo una è effettivamente in vendita, essendo l'altra relativa all'edizione originale del disco, ma entrambe condividono la prima parte dell'URL. La seconda parte, collegata alla prima dalla stringa `/dp/` identifica in modo univoco il prodotto presentato nella pagina. Un permalink completo avrà quindi la forma

`http://www.amazon.com/Songs-Wood-Jethro-Tull/dp/B00008G9JN` (2.1)

mentre è possibile giungere al prodotto anche attraverso il solo secondo codice attraverso link del tipo

`http://www.amazon.com/dp/B00008G9JN` . (2.2)

Come sarà approfondito in seguito, nel database sono presenti i commenti relativi ad un solo prodotto, anche nel caso siano presenti più dischi con la prima parte dell'URL identica: In pratica, sono presenti i commenti relativi ad un solo prodotto (o identificativo del prodotto). Nel database, inoltre, è salvato come identificativo del prodotto anche l'URL intero nella forma (2.1).

2.2.2 Track Listings

La lista delle tracce presenti nell'album è presente, come appena visto, in forma breve nella pagina principale relativa al prodotto e in forma estesa in una pagina dedicata. La tabella contenente la lista della tracce può presentarsi in due forme:

- nel caso le tracce dell'album siano in vendita, in blocco o singolarmente, esse sono presentate in una tabella interattiva dove l'aquirente può ascoltare 30 secondi di anteprima per brano o passare all'aquisto. È questo il caso di Figura 2.1.
- nel caso sia disponibile solo la lista delle tracce, ma i file non siano in vendita, è semplicemente presente una lista delle tracce.

[29] Dovrebbe essere chiaro che il fatto che esistano due differenti versioni di un URL non invalida quanto appena detto riguarda l'unicità degli identificatori dei prodotti. La necessità di evidenziare la presenza di due differenti tipologie di link nasce dal fatto che amazon fornisce le due versioni indifferentemente: ciò che differisce nelle due versioni è la posizione dell'identificavivo all'interno dell'URL e non l'identificativo stesso.

Nonostante le tabelle, nelle due forme, siano piuttosto differenti, è possibile ricavare i nomi delle singole tracce in ogni caso: formalmente, si tratta di unire due parser differenti (uno relativo al primo caso, uno relativo al secondo) facendo affidamento sul fatto che una sola delle due tipologie di tabelle sia presente nella pagina analizzata.

URL L'URL relativo alla pagina dedicata ai *Track Listings* è ottenibile a partire da (2.1) o (2.2) indifferentemente, sostituendo la stringa "`dp/tracks`" alla corrispondente stringa "`dp`".

2.2.3 Commenti

Nonostante, come già accennato, alcuni dei commenti siano presenti nella pagina principale del prodotto, ognuno di essi è associato ad un identificativo univoco che lo rende completamente indipendente e raggiungibile anche singolarmente. Naturalmente, conoscendo l'URL della pagina principale del prodotto, è anche possibile raggiungere, tramite un URL da esso ricavato la cui struttura verrà analizzata in seguito, la lista completa dei commenti al prodotto.

I commenti hanno una formattazione standard piuttosto semplice, presentata in Figura 2.2. Nell'ordine con il quale compaiono, gli elementi che

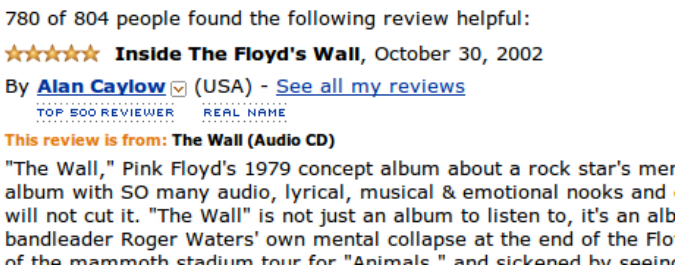
780 of 804 people found the following review helpful:
Inside The Floyd's Wall, October 30, 2002
By Alan Caylow (USA) - See all my reviews
TOP 500 REVIEWER REAL NAME
This review is from: The Wall (Audio CD)
"The Wall," Pink Floyd's 1979 concept album about a rock star's mer
album with SO many audio, lyrical, musical & emotional nooks and
will not cut it. "The Wall" is not just an album to listen to, it's an alb
bandleader Roger Waters' own mental collapse at the end of the Flo
of the mammoth stadium tour for "Animals " and sickened by seein

Figura 2.2: *Screenshot* di un frammento di commento su Amazon.com

compongono un commento sono:

1. La valutazione al commento da parte di altri utenti, sotto forma di risposta binaria alla domanda "Questo commento ti è stato utile?". Questa informazione viene presentata nella forma "*780 of 804 people found the following review helpful*", che tradotta significa "780 persone su 804 hanno trovato utile questo commento". Una semplice sottrazione fornisce il numero di utenti che hanno trovato utile il commento e quanti invece l'hanno trovato inutile, ad esempio in questo caso 24 utenti hanno trovato inutile questo commento, 780 ne sono stati aiutati in qualche modo.

2. La valutazione del prodotto da parte dell'utente, sotto forma di gradimento numerico con una scala da 0 a 5, presentato visivamente sotto forma di stelle colorate.

3. Il titolo del commento, in grassetto in figura.

4. La data di pubblicazione del commento.

5. Il nome dell'autore, il suo nickname tra parentesi e la sua provenienza[30].

6. Subito sotto il nome dell'autore sono presenti, per alcuni autori, delle etichette (o *badge*) che indicano caratteristiche specifiche dell'autore, come essere uno dei 500 migliori *reviwer* (scrittori di recensioni), o presentare il proprio nome reale come nome utente (aumento dell'affidabilità del commento). In questo caso l'autore del commento è ad esempio uno dei 500 migliori *reviewer* di Amazon.com.

URL La lista dei commenti relativi ad un prodotto è ottenibile sempre a partire da (2.1) o (2.2) sostituendo "`product-reviews`" alla stringa "`dp`". È altresì disponibile un URL per ogni singolo commento, la cui struttura è

```
http://www.amazon.com/review/R1YUXAAG4IVURH        (2.3)
```

dove la stringa alfanumerica finale identifica univocamente ogni singolo commento.

2.2.4 Profilo utente

Anche ad ogni utente è associato un identificativo univoco, utilizzando il quale è possibile raggiungere la corrispondente pagina profilo, dalla quale è possibile esplorare ogni recensione e commento redatti dall'utente unitamente ad altre informazioni di contesto spesso fornite da quest'ultimo (in Figura 2.3, ad esempio, sono visibili *Location* ed *E-mail*).

A parte i *badge*, presenti anche in questo caso, è interessante notare che sono disponibili alcune informazioni numeriche riguardo l'utente, alcune sotto forma di percentuale, alcune sotto forma di interi.

Helpful votes received on reviews, il cui valore è visibile in Figura 2.3 subito sotto la foto di Bezos—CEO e fondatore di Amazon.com—è naturalmente la percentuale di voti positivi ricevuti dai commenti scritti dall'utente. I voti positivi di tutti i commenti vengono sommati, così come quelli negativi, e il numero (in arancione, in basso a sinistra) indica la percentuale di voti positivi ricevuti sul totale. *New Reviewer Rank* (insieme al *Classic*) sono invece due indici—o meglio lo stesso indice, recentemente aggiornato

[30] La provenienza è segnalata dall'autore stesso e non scelta in modo rigoroso tra possibilità precise. A ragion di logica può essere falsa.

Figura 2.3: *Screenshot* di una parte di pagina profilo su Amazon.com

e smembrato in due differenti versioni—della qualità delle recensioni di un utente.

Essendo questi indici relativi all'intera produzione di commenti nelle più disparate sezioni della piattaforma, si è deciso di non includerli nel *test set* in corrispondenza delle altre informazioni sui redattori dei commenti. È possibile comunque, come si vedrà in seguito, ricavare un indice simile all'*Helpful votes received* sulle sole recensioni immagazzinate nel *set*, essendo salvate per ogni commento anche le valutazioni degli altri utenti.

2.3 Database: schema concettuale

Nei precedenti paragrafi sono state presentate le pagine web utili alla compilazione del database proposto, ognuna delle quali rappresenta—nella sua totalità o nei singoli elementi che la compongono—una entità. Per procedere all'illustrazione di uno schema concettuale è utile fare riferimento ad un diagrama chiamato *Entity-Relationship Diagram* (in breve ERD), che fa uso dei concetti di entità e relazioni (o associazioni) per illustrare graficamente tale modello che si sta progettando. Un'"entità", in questo caso, rappresenta una classe di oggetti con attributi comuni ed esistenza autonoma: nel nostro caso, ad esempio, il commento è un'entità. Il particolare commento cui fa riferimento l'URI (2.3) viene chiamato "occorrenza" dell'entità commento. Tali entità sono legate le une alle altre da "associazioni" (o "relazioni") tra le entità.

L'ERD mostra anche tutti gli attributi relativi ad ogni entità—un attributo dell'entità commento è ad esempio la sua data—e le cardinalità di ogni relazione: tali cardinalità vengono specificate per ogni entità in una associazione e dicono quante volte in tale associazione un'occorrenza di una di queste entità può essere legata ad occorrenze delle altre entità coinvolte

nell'associazione. Il diagramma che illustra entità e associazioni viste nella sezione precedente è presentato in Figura 2.5: tale schema rappresenta un primo passo nella progettazione del *test set*, che verrà conclusa con la compilazione del modello relazionale nella prossima sezione.

2.3.1 Compilazione del diagramma ER

Risulta immediato tracciare un parallelo tra le varie tipologie di pagine presenti all'interno di Amazon.com analizzate nella sezione precedente e altrettante entità utilizzabili per compilare un diagramma ER completo: l'entità ALBUM rappresenta ogni prodotto di tipo *Audio CD*, mentre SONG fa riferimento alle singole tracce in essi presenti; l'entità ARTIST rappresenta ognuno degli artisti associati agli album; AUTHOR e BADGE fanno riferimento agli utenti della piattaforma ed ai relativi badge; COMMENT rappresenta infine i singoli commenti lasciati dagli utenti.

Album

L'analisi della precedente sezione è partita dall'entità album, indicata con l'etichetta ALBUM in Figura 2.5, al centro. Il suo attributo id identifica l'occorrenza dell'entità all'interno del database, mentre l'attributo amazon_link fa riferimento all'URL presso il quale è possibile fisicamente visualizzare la pagina relativa al prodotto su Amazon.com; in caso di necessità, questo attributo può fornire l'identificativo univoco associato da amazon al prodotto stesso, come visto nella sezione 2.2.3.

Artist

Ad ogni ALBUM è associato un unico artista, rappresentato dall'entità ARTIST attraverso la relazione *hasArtist*, mentre ad un unico ARTIST possono essere associati—tale artista può cioè aver suonato o composto—diversi ALBUM. Ad ogni occorrenza dell'entità ARTIST è associato un identificativo univoco per mezzo dell'attributo id; il nome invece è salvato come attributo name.

Song

Ad ogni ALBUM è anche associata una lista di canzoni, le entità delle quali sono chiamate SONG, attraverso la relazione *hasSong*. Ogni occorrenza di SONG presenta un identificativo e un nome corrispondenti agli attributi id e name che svolgono un ruolo analogo e corrispondente a quanto visto per l'entità ARTIST; è inoltre necessario tenere conto del fatto che una stessa canzone (di uno stesso artista) può comparire in diversi suoi lavori.

Comment

La terza e ultima relazione che si dirama dall'entità ALBUM è *hasComment*, che associa quest'ultima entità all'entità COMMENT, che identifica la classe di commenti di Amazon.com. Gli attributi di questa entità sono quelli evidenziati nella sezione 2.2.3: pos_thumbs e neg_thumbs sono le valutazioni al commento da parte degli altri utenti, come spiegato al punto 1; rating è la valutazione del prodotto da parte dell'utente (in pratica, la valutazione relativa all'occorrenza dell'entità ALBUM associata) come al punto 2; date è la data cui si fa riferimento al punto 4, title è il titolo del commento—punto 3—mentre comment rappresenta il commento stesso. Gli attributi id e amazon_id rappresentano rispettivamente l'identificativo dell'occorrenza all'interno del database e all'interno della piattaforma Amazon: da quest'ultimo parametro è possibile risalire ad un URI per ogni singolo commento come spiegato in 2.2.3.

Author

L'autore di ogni commento è rappresentato dall'entità AUTHOR, associata all'entità COMMENT tramite la relazione *hasAuthor*. L'entità AUTHOR racchiude, oltre agli identificativi univoci del database e di Amazon, i tre attributi descritti al punto 5 della sezione 2.2.3. L'attributo name definisce il nome dell'utente, nickname il suo nome fittizio, o "nome utente", e location il luogo di residenza da lui impostato.

Badge

Ad ogni autore possono essere associati uno o più *badge*, ognuno dei quali può essere visto come un'occorrenza dell'entità BADGE. Quest'ultima entità ha attributi identici alle entità SONG e ARTIST.

2.4 Database: modello relazionale

Prima di procedere alla progettazione logica, è indispensabile introdurre alcuni nuovi concetti e approfondirne alcuni visti nella precedente sezione.

Mentre nel contesto dei diagrammi ER il concetto di relazione (o associazione, termine che sarà usato in vece del primo da qui in poi) è stato introdotto e definito come legame tra differenti entità, nel modello relazionale una relazione è definita, dati gli insiemi $D_1, D_2, \ldots, D_n$, come un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \cdots \times D_n$. È possibile associare ad ogni dominio D_i, all'interno di una relazione, un attributo che indichi il ruolo dell'insieme nella relazione. Lo schema di una relazione consiste di un simbolo R—solitamente una stringa—detto nome della relazione, e un insieme di attributi. Lo schema corrispondente alla relazione AUTHOR, la cui entità

associata è stata introdotta nel capitolo precedente, è illustrabile ad esempio come

AUTHOR (id, amazon_id, name, nickname, location)

dove la sottolineatura evidenzia la chiave primaria. Qui come nel resto del capitolo abbiamo usato e useremo la notazione proposta in Atzeni *e altri* (2002).

È necessario definire anche il concetto di 'vincolo di integrità referenziale' (nei testi anglosassoni e nel linguaggio tecnico *foreign key*), un vincolo inter-relazionale fra un insieme di attributi X di una relazione R_1 e una relazione R_2. Tale relazione è soddisfatta solo se i valori su X di ciascuna tupla di R_1 (o meglio dell'istanza di R_1) compaiono come chiavi di (dell'istanza di) R_2.

Il diagramma ER presentato a pagina 27 può essere trasformato attraverso regole precise, a seconda della molteplicità delle relazioni tra entità, in uno schema logico complesso: questo schema rappresenta le tabelle—e le relazioni tra di esse—realmente implementate nel *test set* ed è rappresentato in Figura 2.4. La notazione utilizzata in questo grafico richiama quella appena illustrata per gli schemi delle relazioni. Le frecce indicano i vincoli di integrità referenziale tra attributi e relazioni, e saranno illustrati nel seguito della sezione.

2.4.1 Da ERD a modello relazionale

Come già accennato nel precedente paragrafo, è possibile convertire in modo formale un diagramma *Entity-Relationship* in uno schema logico. I dettagli riguardo tale conversione non sono argomento di questo elaborato e verranno perciò solo accennati per quanto riguarda le associazioni uno a molti e molti a molti, le uniche presenti nel diagramma ER che ci proponiamo di convertire.

Per quanto riguarda le associazioni molti a molti, la traduzione in modello relazionale di entità e relazioni del diagramma ER prevede

- per ogni entità, una relazione con lo stesso nome avente per attributi gli stessi attributi e per chiave il suo identificatore.
- per l'associazione, una relazione con lo stesso nome avente per attributi gli attributi della associazione e gli identificatori delle entità coinvolte; la chiave per quest'ultima relazione è formata da questi ultimi identificatori.

Per quanto riguarda le associazioni uno a molti invece, seguendo la stessa politica per la traduzione si dovrebbe scrivere una relazione che mutui la propria chiave solo da una delle due entità di origine. Tale relazione 'accessoria' può essere pertanto assorbita nella relazione con la quale condivide la chiave.

È possibile ora proseguire con la traduzione dello schema concettuale di pagina 27 in quello logico di pagina 25. La traduzione è stata divisa in due

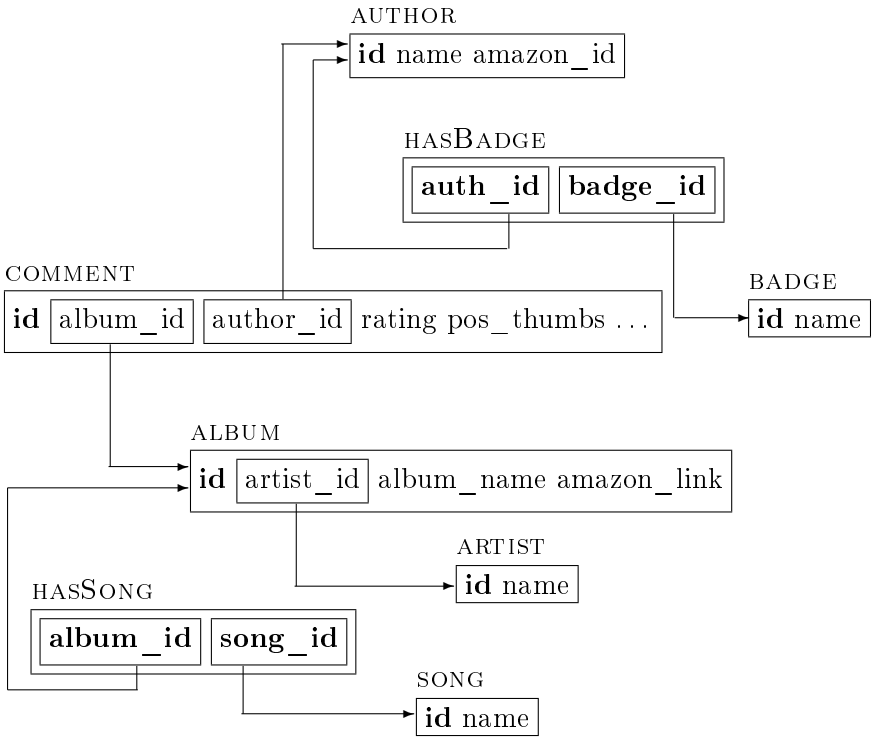

Figura 2.4: Illustrazione grafica del modello relazionale corrispondente al diagramma ER di Figura 2.5

parti per chiarezza e semplicità e non entra né nei dettagli della traduzione, che è possibile approfondire tramite il testo già citato (Atzeni *e altri*, 2002), né nei dettagli puramente implementativi, trattati separatamente nel Capitolo 3 più avanti.

Album, Song, Artist

La presenza dell'associazione *hasSong*, molti a molti, dall'entità ALBUM verso l'entità SONG rende necessaria, come appena visto, la presenza di una tabella HASSONG che mantenga i riferimenti verso le tabelle ALBUM e SONG stesse. Per implementare l'associazione *hasAuthor*, un attributo deve invece essere aggiunto a quelli 'ereditati 'da ALBUM a partire dal diagramma stesso. Gli schemi delle tre relazioni (nella seconda accezione, già largamente utilizzata in questo capitolo) risultanti da tale elaborazione, insieme alla quarta

relazione aggiunta, saranno quindi rappresentabili come

ALBUM (id, amazon_link, album_name, author_id)
ARTIST (id, name)
SONG (id, name)
HASSONG (song_id, album_id)

secondo la convenzione già illustrata.

Gli identificatori univoci delle prime tre tabelle sono stati mutuati da quelli scelti durante la compilazione del diagramma ER; per la relazione HASSONG invece la chiave è composta dagli identificatori delle entità coinvolte, come accennato precedentemente. Tale situazione è chiaramente rappresentata sia negli schemi di relazione qui sopra sia che in Figura 2.4: nei primi è evidenziata dal fatto che entrambi gli attributi sono sottolineati, nella seconda dal fatto che entrambi gli attributi sono evidenziati in grassetto.

Nel passaggio da diagramma ER a schema relazionale, gli attributi aggiunti a quelli originali di ogni entità sono sottoposti a vincoli di integrità referenziale. Tali vincoli non sono evidenti negli schemi delle relazioni qui sopra ma gli attributi author_id di ALBUM e song_id e album_id di HASSONG sono vincolati come evidentemente espresso in figura: i dettagli riguardo tali vincoli sono argomento della sezione 3.1.1.

Comment, Author, Badge

Anche per le rimanenti entità e associazioni utilizziamo il metodo illustrato all'inizio della sezione ed utilizzato fino ad ora. La relazione *hasComment*, presente nel diagramma *Entity-Relationship* a collegare le entità COMMENT ed ALBUM—uno a molti—viene assorbita come attributo dalla tabella corrispondente alla prima entità

COMMENT (id, item_id, author_id, pos_thumbs, neg_thumbs, date, comment ...)

dove i puntini di sospensione stanno ad indicare alcuni attributi omessi per questioni di spazio. L'associazione *hasBadge* tra AUTHOR e BADGE è del tutto analoga alla relazione *hasSong* vista prima e per essa valgono tutte le considerazioni fatte per quest'ultima. Gli schemi per le restanti relazioni possono essere quindi rappresentati come

AUTHOR (id, name amazon_id)
HASBADGE (auth_id, badge_id)
BADGE (id, name)

mentre i vincoli sono ancora una volta evidenti in figura.

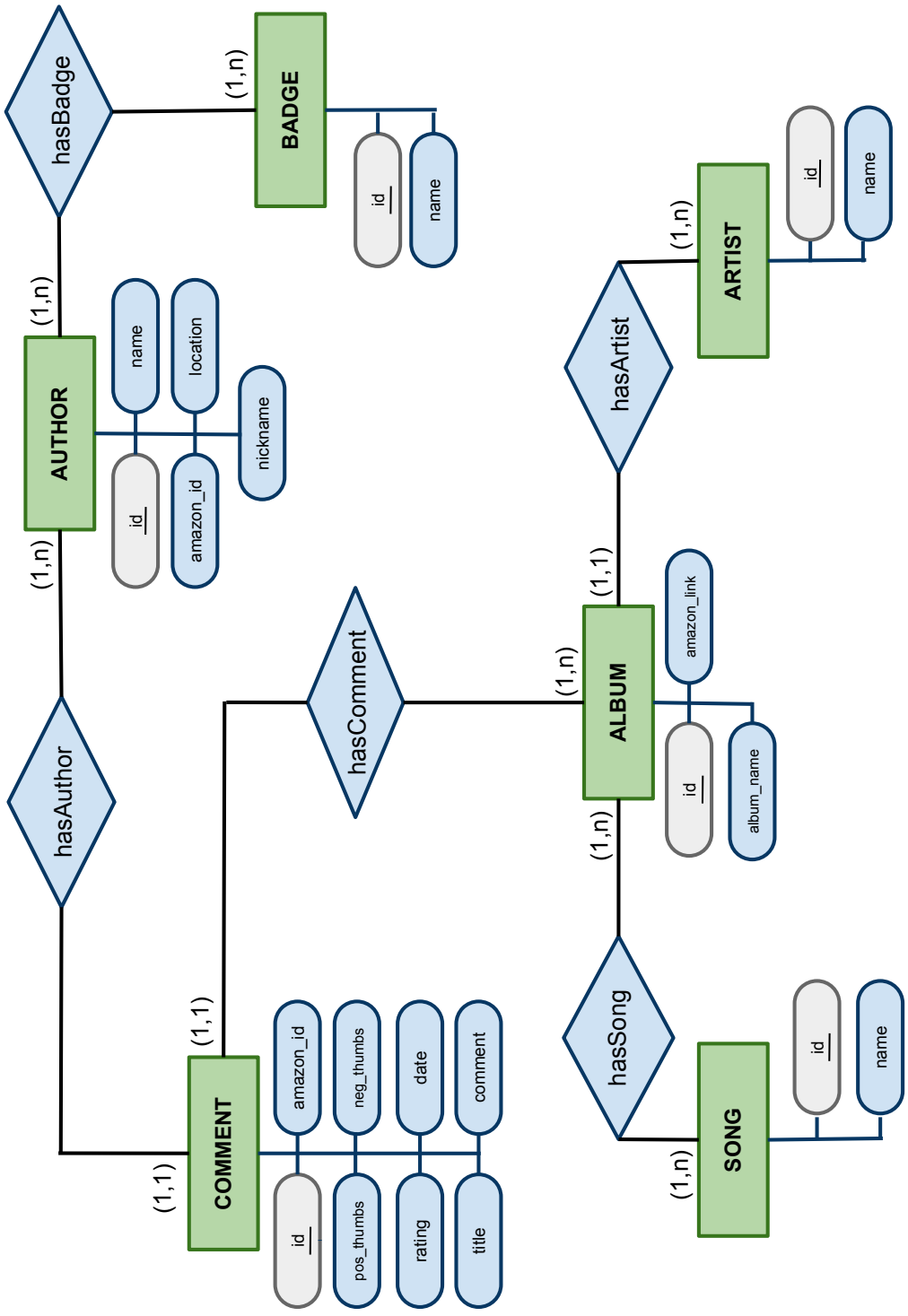

Figura 2.5: ERD per entità e relazioni analizzate.

Capitolo 3

Implementazione

Nel capitolo precedente abbiamo introdotto la progettazione concettuale e il passaggio al modello relazionale del *test set* dopo aver illustrato la struttra di Amazon.com e i vincoli dettati da quest'ultima. In questo capitolo concluderemo la descrizione dell'implementazione del database; verranno inoltre evidenziate le tecnologie utilizzate per la memorizzazione fisica dei dati e per l'importazione/esportazione degli stessi.

3.1 Il database

In questa sezioni illustreremo l'implementazione del *database*, sulla base dei modelli ricavati nel precedente capitolo, dove è stata illustrata in modo piuttosto dettagliato la creazione dello schema concettuale e la sua traduzione in modello relazionale.Il sistema scelto per la gestione del database (o meglio il *object-relational database management system* scelto) è stato PostgreSQL[1]. La scelta è caduta su questo sistema innanzi tutto perchè è rilasicato con licenza MIT, ed è quindi gratuito ed *open-source*; come molti software di questo tipo inoltre, PostgreSQL non è controllato da una singola azienda ma da una community di sviluppatori, ed esiste un'ampia base di utenti pronti ad aiutare e fornire supporto reciproco.

Essendo già stato introdotto il modello relazionale e il relativo schema (Figura 2.4 a pagina 25) ed essendo chiara la scelta del sistema di gestione del database utilizzato, dovrebbe essere piuttosto immediato comprendere il listato in Figura 3.1 a pagina 34, nel quale è presentato il codice utilizzato per la creazione del database. Piuttosto che proseguire con un'analisi metodica di ogni tabella, nei prossimi paragrafi analizzeremo le scelte fatte riguardo l'implementazione dell'intero database, senza soffermarci su ogni attributo.

[1] `http://www.postgresql.org`

3.1.1 Vincoli di integrità referenziale

Il *test set* è composto da otto tabelle, e le relazioni tra tali tabelle sono state illustrate come vincoli di integrità referenziale (definiti nella sezione 2.4), senza entrare in particolari dettagli sulla natura di tali *foreign key* e sulla risposta alle violazioni: per quanto riguarda queste impostazioni, il database è del tutto omogeneo ed è più opportuno illustrare una sola volta il suo funzionamento. Tutte le *foreign key* presenti sono impostate a `CASCADE` sia `ON DELETE` che `ON UPDATE`: nel caso un vincolo venga cioè a cadere nella tabella esterna a causa di una modifica nella stessa (nei termini utilizzati all'inizio della sezione 2.4, una modifica ad un'istanza della relazione R_2), tale modifica si ripercuote direttamente sul corrispondente attributo della tabella interna (un'istanza di R_1). Questa scelta è stata portata avanti pensando all'uso che la base di dati costruita dovrebbe o potrebbe avere in futuro, ossia quella di esplorare, indagare, analizzare una grande quantità di informazioni coerenti, la quale validità e compostezza sia garantita e non quello di immagazzinare nuovi dati.

In un uso che preveda la sola consultazione della base dati, non dovrebbe in ogni caso essere possibile modificare un record di un'istanza di nessuna delle relazioni presenti; anche se questo dovesse succedere, adottanto la politica di propagare le modifiche a tutte le tabelle collegate, il *set* rimarrebbe comunque coerente e utilizzabile.

3.1.2 Identificatori e chiavi

Nella sezione 2.4.1 e nel diagramma di Figura 2.4 sono evidenziate in modo chiaro le chiavi primarie scelte per ogi relazione, mentre all'inizio della sezione 2.4.1 è stata illustrata la differenza tra le chiavi mutuate da quelle presenti nel diagramma ER (sottolineate in Figura 2.5) e quelle venutesi a creare a causa di associazioni molti a molti nella traduzione tra ERD e modello relazionale.

Nel primo caso, ad esempio per quanto riguarda le relazioni AUTHOR o ARTIST, l'id è specificato come `SERIAL PRIMARY KEY`, il che garantisce l'unicità della chiave e la sua identificazione all'interno della base di dati come valore che si incrementa in modo automatico ad ogni inserimento. Nel secondo caso, ad esempio per le relazioni HASSONG o HASBADGE, per i due attributi che fanno anche da chiavi non è necessario alcun vincolo oltre quello di FOREIGN KEY già discusso nella sezione precedente.

3.1.3 Attributi nulli e unici

Tutti gli attributi, a parte nickname e location nella relazione AUTHOR, sono dichiarati `NOT NULL` e nessuna loro istanza può quindi assumere valore nullo. I due attributi evidenziati possono essere nulli perchè sono parametri scelti dagli utenti e possono variare di caso in caso. Oltre gli identificatori interni

al database, anche gli id ricavati da Amazon sono dichiarati `UNIQUE`, così come gli attributi name di BADGE e ARTIST.

Tutti i valori presenti nel database sono stati ricavati da Amazon.com con le modalità che saranno brevemente spiegate nella sezione 3.3; il fatto che nessun attributo possa assumere valore nullo è conseguenza immediata di questo fatto: se un record è stato immesso nel database, le informazioni ad esso corrispondente erano presenti sulla piattaforma e non saranno certamente nulle! La scelta di dichiarare unici gli attributi citati sopra è stata invece più problematica e fonte di discussione: oltre ad aver aiutato a controllare la validità dei dati progressivamente inseriti nella base dati evitando sistematicamente l'inserimento erroneo di doppioni, però, questa dichiarazione fornisce anche una garanzia nei confronti degli utilizzatori del TEST SET sulla robustezza del database ed è stata perciò mantenuta.

3.1.4 Il problema dell'entità song

Durante l'analisi del diagramma ER, abbiamo descritto l'entità SONG come corrispondente ad ogni canzone presente in ognuno degli ALBUM. Quest'ultima entità è collegata alla prima attraverso la relazione *hasSong*, e gli attributi di SONG sono semplicemente id e nome, name: il relativo artista può essere ricavato tramite l'associazione *hasSong* e l'artist_id di ALBUM. Durante l'inserimento dei dati, però, per facilitare il riconoscimento ed evitare l'inserimento di doppioni si è mantenuto anche un riferimento diretto all'id dell'artista di riferimento per la canzone all'interno della relazione SONG (senza quindi dover passare per una serie di laboriose—anche in termini temporali—richieste al database). Questo riferimento è del tutto inutile se si considera il modello concettuale del database, ma proprio a causa della frequenza con la quale dati relativi artista e traccia sono richiesti contemporaneamente questo riferimento è stato mantenuto e se ne trova un'evidente traccia nel listato di Figura 3.1.

3.2 I dati ottenuti

Le modalità utilizzate per recuperare i dati e per popolare il database saranno discusse nella sezione successiva, mentre in questa ci concentreremo su una breve analisi dei dati ottenuti. Come già spiegato in precedenza, unire alla raccolta dei dati un'analisi approfondita degli stessi insieme ad una valutazione, ad esempio, di un sistema MIR in uso è prerogativa di una ricerca più approfondita di quanto fosse possibile fare in questo elaborato.

A partire dai 500 album, scelti utilizzando la classifica dei 500 migliori album secondo la rivista Rolling Stones[2], sono state trovate 497 corrispon-

[2] 500 Greatest Albums by Rolling Stones Magazine
`http://www.rollingstone.com/music/lists/500-greatest-albums-of-all-time-19691231`

denze su Amazon.com sotto forma di album firmati da 277 artisti differenti. I commenti scaricati ed immagazzinati in totale sono 94276, scritti da 41346 diversi autori che condividono 9 *badge* differenti.

Un primo dato assolutamente evidente è la presenza di un numero molto inferiore ai 500 artisti: si pensi soltanto che la band "The Beatles" vanta ben 10 dischi all'interno della classifica. È importante notare, comunque, che le basi su cui poggiano la maggior parte dei moderni sistemi di *Information Retrieval* sono le singole tracce più che interi album; è evidente che questa porzione relativamente ristretta di artisti copre in realtà una porzione molto ampia degli ascolti, nonostante il campione non sia statistico ma (quasi) completamente arbitrario[3].

Un secondo dato che è invece certamente incoraggiante è la presenza di un numero di utenti molto minore rispetto al numero dei commenti. Durante la fase di valutazione dei requisiti non è sempre stata data per scontata la scelta di registrare in modo così preciso gli utenti (nel *set* l'entità AUTHOR) e i badge ad essi associati. La scelta di indirizzare l'intero elaborato alla compilazione di un database utile e riutilizzabile ha però portato a questa scelta che si è rivelata fruttuosa: oltre ad analizzare la relazione tra valutazioni diverse all'interno dello stesso commento (numerica e testuale, ad esempio), potrebbero ad esempio essere utilizzate le informazioni riguardo gli utenti per ottenere dati ancora più precisi sui singoli commenti, calcolando ad esempio l'affidabilità di un utente tramite il calcolo degli indici discussi nella sezione 2.2.4.

3.3 Costruzione ed esportazione del database

Nella precedente sezione sono illustrati i dettagli implementativi del *test set*, mentre abbiamo sorvolato tutte le informazioni riguardo la sua gestione, sia per quanto concerne sua creazione si riguardo una sua eventuale esportazione ed un suo utilizzo. Queste due parti verranno illustrate separatamente nel corso dei prossimi paragrafi.

Creazione del database, importazione dati Per recuperare i dati da immagazzinare è stato costruito un rudimentale *parser* HTML programmato per l'analisi delle pagine web i quali indirizzi sono stati descritti in dettagli nel Capitolo 2. Il processo del *parsing*, o *analisi sintattica*, nel linguaggio informatico denota il processo di analisi di un testo per determinarne la struttura grammaticale con riferimento ad una particolare grammatica—in questo caso formata dai tag HTML. Il *parser* è il *software* che svolge questo compito.

[3] Si pensi all'ormai celebre teoria di Chris Anderson riguardo la coda lunga, "The Long Tail", della quale è facile capire le implicazioni nel mondo del mercato musicale (Celma, 2008).

Il linguaggio utilizzato è stato Java, mentre per riversare i dati ottenuti dal *parser* verso il database—PostgreSQL come già spiegato—abbiamo fatto uso delle librerie JCDB (*Java DataBase Connectivity*), delle API per Java che hanno permesso di interagire in maniera molto semplice con il database.

Esportazione ed utilizzo Una volta completata la preparazione del database, abbiamo concentrato la nostra ricerca sul come distribuire il *test set* in modo semplice, più immediato possibile. Fortunatamente, PostgreSQL stesso offre tutti gli strumenti necessari a svolgere questo compito nel migliore dei modi.

Il database completo è stato esportato utilizzando il comando `pg_dump` con l'opzione `-Fc`, che crea un file con estensione personalizzata, compresso di default, attraverso il quale è poi possibile, tra l'altro, importare anche solo parti del database originale. Una volta ottenuto tale file è possibile importare l'intero *test set* utilizzando il comando `pg_restore` con la sintassi

`pg_restore -U` UTENTE `--no-owner -d` NOME_DB FILE

dove UTENTE è il nome utente del database NOME_DB che si sta utilizzando mentre FILE è il file del *test set* del quale si è parlato qualche riga sopra. Questo comando crea una copia esatta del database trasferendo tutti i diritti all'utente specificato.

```
1  CREATE TABLE author (
2     id SERIAL PRIMARY KEY,
3     amazon_id text UNIQUE,
4     name TEXT NOT NULL,
5     nickname TEXT,
6     location TEXT );
7
8  CREATE TABLE artist (
9     id SERIAL PRIMARY KEY,
10    name TEXT UNIQUE NOT NULL );
11
12 CREATE TABLE album (
13    id SERIAL PRIMARY KEY,
14    amazon_link TEXT UNIQUE NOT NULL,
15    album_name TEXT NOT NULL,
16    artist_id INT NOT NULL REFERENCES artist(ID) ON DELETE CASCADE
          ON UPDATE CASCADE );
17
18 CREATE TABLE comment (
19    id SERIAL PRIMARY KEY,
20    amazon_id TEXT UNIQUE NOT NULL,
21    pos_thumbs INT NOT NULL,
22    neg_thumbs INT NOT NULL,
23    rating DOUBLE PRECISION NOT NULL,
24    date DATE NOT NULL,
25    title TEXT NOT NULL,
26    comment TEXT NOT NULL,
27    location TEXT NOT NULL,
28    auth_id INT NOT NULL REFERENCES author(ID) ON DELETE CASCADE
          ON UPDATE CASCADE,
29    album_id INT NOT NULL REFERENCES album(ID) ON DELETE CASCADE
          ON UPDATE CASCADE );
30
31 CREATE TABLE badge (
32    id SERIAL PRIMARY KEY ,
33    name TEXT UNIQUE NOT NULL );
34
35 CREATE TABLE hasbadge (
36    badge_id INT NOT NULL REFERENCES badge(ID) ON DELETE CASCADE
          ON UPDATE CASCADE,
37    auth_id INT NOT NULL REFERENCES author(ID) ON DELETE CASCADE
          ON UPDATE CASCADE );
38
39 CREATE TABLE song (
40    id SERIAL PRIMARY KEY,
41    name TEXT NOT NULL,
42    artist TEXT NOT NULL REFERENCES artist(ID) ON DELETE CASCADE
          ON UPDATE CASCADE );
43
44 CREATE TABLE hassong (
45    album_id INT NOT NULL REFERENCES album(ID) ON DELETE CASCADE
          ON UPDATE CASCADE,
46    song_id INT NOT NULL REFERENCES song(ID) ON DELETE CASCADE ON
          UPDATE CASCADE );
```

Figura 3.1: Codice SQL per la creazione del *test set* ricavato dal modello relazionale di Figura 2.4

Conclusioni

L'oggetto di questo elaborato è stata la progettazione e la creazione di una base di dati, recuperati da Amazon.com, per la valutazione di motori di ricerca musicali ma in ogni caso aperta ad ogni utilizzo se ne desideri fare. La creazione del *test set* è stata preceduta dall'analisi di metodi e tecnologie usati nei sistemi di IR e di MIR e dalla presentazione di alcune piattaforme di questo tipo. Le fasi principali del progetto sono state:

- studio delle pubblicazioni sull'argomento, studio delle piattaforme più conosciute di *Music Information Retrieval* e delle loro API;
- scelta dei dati per la creazione del database, analisi della struttura di Amazon.com in particolare;
- progettazione del database, implementazione dello stesso;
- progettazione ed implementazione del parser HTML per Amazon.com, riversamento dei dati nel database.

La prima fase del lavoro ha riguardato lo studio della letteratura riguardante *Information Retrieval* e *Music Information Retrieval*, della quale si possono trovare interessanti spunti nella Bibliografia, per approfondire soprattutto i concetti basilari di queste discipline, i metodi utilizzati più di frequente e i più comuni problemi. A seguito di questo abbiamo deciso di indirizzare la nostra ricerca verso l'ambito dello studio delle informazioni di contesto più che del contenuto audio delle tracce, un approccio molto studiato che sta progressivamente lasciando il posto anche alle raccomandazioni basate su quest'ultimo tipo di analisi, ad esempio grazie alla disponibilità di nuove tecnologie aperte agli sviluppatori.

La scelta di Amazon.com come piattaforma dalla quale ricavare dati interessanti è stata stimolata dall'assenza di pubblicazioni, dalla quantità di dati effettivamente recuperabili sull'argomento e dall'interesse suscitato da ricerche simili a questa contemporaneamente portate avanti da altri studenti. La scelta di quali dati effettivamente includere nel database è stata più difficoltosa e ha dimostrato quanto creare una base di dati coerente con un progetto formalmente corretto fin dal principio possa essere complicato. Creare il database nel modo più corretto possibile fin dall'inizio è stata, però, anche la parte più stimolante del progetto stesso.

Mentre la compilazione del software di *parsing* delle pagine di Amazon.com non ha portato particolari problemi, è stato più difficoltoso imparare a far interagire Java e il database SQL preparato. Ottimizzare i tempi per il parsing, il download ed il trasferimento dei dati non è stata una priorità dell'implementazione, ma nel caso si procedesse in futuro ad ampliare il volume del database si renderebbe necessaria una riscrittura del codice.

Come già fatto notare, proseguire nel progetto con l'analisi dei risultati sarebbe stato fuori dalle nostre possibilità, ma tale approfondimento potrebbe certamente portare a risultati interessanti ed un progetto di questo tipo si presenta a tutti gli effetti come possibile prosieguo del lavoro fin qui svolto. Un'altra prospettiva interessante è quella di ampliare il database fino a comprendere informazioni riguardanti una quantità maggiore di tracce, album ed artisti: un'idea potrebbe ad esempio essere quella di ampliare la collezione fino a comprendere gli album per tutte le tracce contenute all'interno del Million Song Dataset, riguardo il quale è possibile trovare un breve riferimento all'inizio del Capitolo 2, che sta diventando un punto di riferimento nel mondo accademico per lo studio di sistemi MIR e non solo.

Bibliografia

Abowd G.; Dey A.; Brown P.; Davies N.; Smith M.; Steggles P. (1999). Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pp. 304–307, London, UK. Springer-Verlag.

Atzeni P.; Ceri S.; Paraboschi S.; Torlone R. (2002). *Basi di dati.* McGraw-Hill.

Baeza-Yates R. A.; Ribeiro-Neto B. (1999). *Modern Information Retrieval.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Barrington L.; Oda R.; Lanckriet G. (2009). Smarter than genius? human evaluation of music recommender systems. In *Proceedings of the international conference on Multimedia information retrieval*, pp. 357–362.

Benvenuto N.; Zorzi M. (2010). *Principles of Communications Networks and Systems.* John Wiley.

Bogdanov D.; Haro M.; Fuhrmann F.; Xambó A.; Gómez E.; Herrera P. (2011). A content-based system for music recommendation and visualization of user preferences working on semantic notions. In *9th International Workshop on Content-based Multimedia Indexing*, Madrid, Spain.

Breese J. S.; Heckerman D.; Kadie C. M. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14^{th} Conference on Uncertainty in Artificial Intelligence.* A cura di Cooper G. F., Moral S., pp. 43–52.

Celma Ó. (2008). *Music Recommendation and Discovery in the Long Tail.* Tesi di Dottorato di Ricerca, University Pompeu Fabra, Barcelona, Spain.

Lamere P.; Celma Ó. (2007). Music recommendation tutorial notes. *ISMIR Tutorial.*

Meadow C. T.; Boyce R. B.; Kraft H. D. (1992). *Text Information Retrieval Systems.* Academic Press.

Miotto R.; Orio N. (2010). A probabilistic approach to merge context and content information for music retrieval. In *Proceedings of the ISMIR Conference*, pp. 15–20.

Salton G.; McGill M. (1986). *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA.

Sarwar B.; Karypis G.; Konstan J.; Reidl J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pp. 285–295, New York, NY, USA. ACM.

Shannon C. E.; Weaver W. (1959). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois.

Tingle D.; Kim Y. E.; Turnbull D. (2010). Exploring automatic music annotation with acoustically-objective tags. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pp. 55–62, New York, NY, USA. ACM.

Turnbull D.; Barrington L.; Lanckriet G. (2008). Five approaches to collecting tags for music. In *Proceedings of the ISMIR Conference*, pp. 225–230.