Università degli Studi di Salerno

Dipartimento di Informatica

Tesi di Laurea Triennale in

Informatica

UN ALGORITMO PER L'INFERENZA DI DIPENDENZE FUNZIONALI RILASSATE: VERIFICA DI AMMISSIBILITA' DI PATTERN

Relatore

Chiar.mo Prof. Vincenzo Deufemia

Secondo Relatore

Dott.sa Loredana Caruccio

Candidato

Luigi Durso

Matr. 0512101919

Anno Accademico 2016-2017

Alla mia famiglia e alla mia ragazza,

per aver creduto in me e per avermi sostenuto nei momenti più difficili.

Abstract

Le basi di dati sono utilizzate in larga scala in moltissimi aspetti dell'ambito tecnologico, per questo motivo durante la loro progettazione ci sono aspetti essenziali da prendere in considerazione per assicurare un servizio quanto più efficiente possibile. La qualità dei dati contenuti è un servizio che di certo una buona base di dati deve garantire, motivo per il quale la *data quality* è divenuta una materia estremamente interessante negli ultimi anni. Per ridurre anomalie ed inconsistenze ci vengono incontro le *Dipendenze funzionali*, utilizzate ampiamente per stabilire vincoli di integrità tra i dati. La grande mole di dati, però, ha reso necessario un riadattamento delle dipendenze funzionali rendendole in grado di catturare inconsistenze più ampie nei dati. Le *Dipendenze funzionali rilassate o approssimate* **(RFD)** sono da considerarsi come una naturale evoluzione o generalizzazione delle *dipendenze funzionali canoniche*. Il concetto più importante introdotto dalle RFD è quello della *similarità*. Nelle dipendenze funzionali classiche esisteva soltanto il concetto di uguaglianza tra dati, nelle RFD espandiamo questo concetto ad una similarità, questo ci permetterà di coprire una quantità di dati maggiore. Tuttavia le RFD possono fornire vantaggi solo se possono essere scoperte automaticamen-

te. Il lavoro di tesi si è basato su questo ultimo concetto di ottenere le RFD in seguito ad una procedura automatizzata. Durante le varie fasi di studio si è pensato ed implementato un algoritmo che permette, attraverso tre fasi intermedie, la scoperta di RFD di un dataset dato come input. Per questo lavoro di tesi mostreremo l'idea dell'algoritmo generale ed entreremo nel dettaglio della prima fase di sviluppo(Feasibility), mostrando, infine, i risultati della sperimentazione.

Indice

Elenco delle tabelle

Snippet di codice

Introduzione

1.1 Incipit

Nella progettazione di una base di dati ci sono aspetti essenziali da prendere in considerazione per assicurare un servizio quanto più efficiente possibile. Uno di questi servizi è certamente la *qualità dei dati*, una base di dati con questa caratteristica farà si che le inconsistenze tra i dati siano il minor numero possibile. Negli ultimi anni la crescita delle reti ha portato ad un aumento considerevole del flusso di dati rendendo la *data quality* una materia estremamente interessante vista la cospicua presenza di dati "sporchi" proveniente da fonti differenti. Per ridurre questo tipo di anomalie è impensabile tentare di eliminare le *inconsistenze* manualmente, una procedura di questo tipo può essere facilmente incline ad errori sopratutto con la quantità di dati precedentemente citata. In questo lavoro ci vengono incontro le *Dipendenze funzionali*, utilizzate ampiamente per stabilire vincoli di integrità tra i dati e ridurre anomalie e inconsistenze all'interno della nostra base di dati. La grande mole di

dati, però, ha reso necessario un riadattamento delle dipendenze funzionali rendendole in grado di catturare inconsistenze più ampie nei dati. Le *Dipendenze funzionali rilassate o approssimate* **(RFD)** sono da considerarsi come una naturale evoluzione o generalizzazione delle *dipendenze funzionali canoniche*. Questo nuovo strumento ci permette di adattare le semplici dipendenze funzionali a diversi contesti applicativi, infatti, le RFD possono applicarsi anche solo ad una porzione di database. Il concetto più importante introdotto dalle RFD, però, è quello della *similarità*. Nelle dipendenze funzionali classiche esisteva soltanto il concetto di uguaglianza tra dati, nelle RFD espandiamo questo concetto ad una similarità, questo ci permetterà di coprire una quantità di dati maggiore e sfruttare le RFD appena scoperte per effettuare una operazione di *cleaning* sulla base di dati. Tuttavia le RFD possono fornire vantaggi solo se possono essere scoperte automaticamente. Il lavoro di tesi si è basato su questo ultimo concetto di ottenere le RFD in seguito ad una procedura automatizzata. Durante le varie fasi di studio si è pensato ed implementato un algoritmo che permette, attraverso tre fasi intermedie, la scoperta di RFD di un dataset dato come input. Le tre fasi di questo algoritmo sono: *Feasibility,Minimality,RFD Discovery* . Per questo lavoro di tesi mostreremo l'idea dell'algoritmo generale ed entreremo nel dettaglio della prima fase di sviluppo(Feasibility), mostrando, infine, i risultati della sperimentazione. Per questo algoritmo, particolare attenzione è stata posta sull'efficienza, oltre che sull'efficacia, studiando un'implementazione basata sul multithreading e predisponendola ad eventuale adattamento parallelo.

1.2 Nozioni Preliminari

E' necessario, prima di cominciare con lo studio del nostro algoritmo, introdurre alcuni concetti preliminari volti alla comprensione della logica dietro le RFD.

1.2.1 Schema di relazione

Uno schema di relazione è costituito da un simbolo R, detto nome della relazione, e da un insieme di attributi $X = \{A_1, A_2, ..., A_n\}$, di solito indicato con $R(X)$. A ciascun attributo $A \in X$ e associato un dominio $dom(A)$. Uno schema di base di dati è un insieme di schemi di relazione con nomi diversi:

$$R = \{R_1(X_1), R_2(X_2), \ldots, R_n(X_n)\}.$$

Una relazione su uno schema $R(X)$ è un insieme r di tuple su X. Per ogni istanza $r \in R(X)$, per ogni tupla $t \in r$ e per ogni attributo $A \in X$, $t[A]$ rappresenta la proiezione di A su t. In modo analogo, dato un insieme di attributi $Y \subseteq X$, $t[Y]$ rappresenta la proiezione di Y su t.[1]

Matricola	Cognome	Nome	Data di nascita
123456	Rossi	Mario	25/11/1991
567891	Neri	Anna	23/04/1992

Tabella 1.1: Esempio di schema di relazione

1.2.2 Dipendenze funzionali canoniche

Una *dipendenza funzionale*, abbreviata in FD, è un vincolo di integrità semantico per il modello relazionale che descrive i legami di tipo funzionale tra gli attributi di una relazione.

Data una relazione r su uno schema $R(X)$ e due sottoinsiemi di attributi non vuoti Y e Z di X, diremo che esiste su r una dipendenza funzionale tra Y e Z, se, per ogni coppia di tuple t_1 e t_2 di r aventi gli stessi valori sugli attributi Y, risulta che t_1 e t_2 hanno gli stessi valori sugli attributi Z:

$$\forall t_1, t_2 \in r, t_1[Y] = t_2[Y] \implies t_1[Z] = t_2[Z] \tag{1.1}$$

Una dipendenza funzionale tra gli attributi Y e Z viene indicata con la notazione $Y \to Z$ e viene associata ad uno schema.

Se l'insieme Z è composto da attributi $A_1, A_2, \ldots, A_k$, allora una relazione soddisfa $Y \to Z$ se e solo se essa soddisfa tutte le k dipendenze $Y \to A_1$, $Y \to A_2$,..., $Y \to A_k$. Di conseguenza, quando opportuno, possiamo assumere che le dipendenze abbiano la forma $Y \to A$, con A singolo attributo.

Una relazione funzionale è *non banale* se A non compare tra gli attributi di Y.

Data una chiave K di una relazione r, si può facilmente notare che esiste una dipendenza funzionale tra K ed ogni altro attributo dello schema di r. Quindi una dipendenza funzionale $Y \to Z$ su uno schema $R(X)$ degenera nel vincolo di chiave se l'unione di Y e Z è pari a X. In tal caso Y è superchiave per lo schema $R(X)$.

Con la notazione $\langle R(X), F \rangle$ indicheremo uno schema $R(X)$ su cui è definito un

insieme di dipendenze funzionali F. Un'istanza r di $R(X)$ viene detta *istanza legale* di $\langle R(X), F\rangle$ se soddisfa tutte le dipendenze funzionali in F. Infine, data una relazione funzionale $Y \to Z$, se ogni istanza legale r di $\langle R(X), F\rangle$ soddisfa anche $Y \to Z$, allora diremo che F *implica logicamente* $Y \to Z$, indicato come $F \models Y \to Z$.

1.2.3 Dipendenze funzionali rilassate

1.2.4 Teoria delle decisioni

1.2.5 Dominanza

1.3 Studi preliminari

Stato dell'arte

Questo è lo stato dell'arte

Bibliografia

[1] F. P. P. S. T. R. Atzeni P., Ceri S., *Basi di dati: Modelli e linguaggi di programmazione*. McGraw Hill, 2013.