



EpMiner:

Scoperta di Pattern Frequenti e Pattern Emergenti

Caso di studio di Metodi Avanzati di
Programmazione AA 2020-2021

Realizzato da:

Lorena Capotorto 696947
l.capotorto@studenti.uniba.it

Roberto Dimatteo 707120
r.dimatteo6@studenti.uniba.it

Sommario

1. Introduzione	2
1.1 Descrizione del Progetto	2
1.2 Algoritmo Apriori.....	3
2. Guida all'Installazione	5
2.1 Installazione server	5
2.2 Installazione client.....	5
3. Guida utente	6
3.1 Applicativo server	6
3.2 Applicativo client.....	6
Note	12

1. Introduzione

Il caso di studio analizzato nell'anno 2020-2021 ha visto come soggetto la scoperta di pattern frequenti e pattern emergenti.

I pattern frequenti sono stati inizialmente definiti nel Market Basket Analysis per scoprire le regolarità di acquisto di un cliente.

Lo scopo è di identificare un insieme di item che occorrono con una frequenza minima ($\geq \text{minSup}$) in D , dove:

- D rappresenta una collezione di transazioni;
- un item è una transazione rappresentata con un vettore di coppie attributo-valore;
- minSup è un valore minimo di supporto che è compreso tra 0 e 1 ($0 < \text{minSup} \leq 1$).

I pattern emergenti descrivono correlazioni tra item la cui forza è in aumento nel tempo.

Lo scopo è trovare tutti i pattern emergenti che siano frequenti (supporto $\geq \text{minSup}$) in DataTarget ed emergenti ($\text{grow rate} \geq \text{minGr}$) rispetto a DataBackground , dove:

- DataTarget è un database di transazioni target;
- DataBackground è un database di transazioni di background;
- minSup è un valore minimo di supporto che è compreso tra 0 e 1 ($0 < \text{minSup} \leq 1$);
- minGr è un valore di minimo grow rate ≥ 1 .

1.1 Descrizione del Progetto

Il software realizzato utilizza l'algoritmo Apriori elaborando dati estratti da una tabella presente in un database di tipo MySQL e consiste in un'applicazione distribuita di tipo Client/Server.

Il server si occupa di mostrare lo stato delle connessioni e di ricevere le richieste di uno o più client, i quali posso effettuare le seguenti operazioni:

- ricerca di un nuovo pattern nel database;
- caricamento di un pattern salvato in archivio nel database;
- uscire dal programma e chiudere la connessione.

In entrambi i casi, il client dovrà specificare nei criteri di ricerca:

- minimo supporto;
- minimo grow rate;
- nome delle tabelle target e background.

L'estensione è stata sviluppata usando la tecnologia JavaFX; inoltre, è stato utilizzato SceneBuilder per la creazione dell'interfaccia grafica e CSS per migliorare la user experience.¹

1.2 Algoritmo Apriori

L'algoritmo Apriori è stato proposto da R. Agrawal e R. Srikant nel 1994.

Esso è progettato per operare su database contenenti transazioni, dove ogni transazione è vista come un insieme di elementi. Data una soglia C , l'algoritmo Apriori identifica gli insiemi di elementi che sono sottoinsiemi di almeno C transazioni nel database.

Apriori utilizza un approccio "dal basso verso l'alto", in cui i sottoinsiemi frequenti vengono estesi un elemento alla volta e i gruppi di candidati vengono testati rispetto ai dati.

L'algoritmo termina quando non vengono trovate ulteriori estensioni riuscite.

Inoltre, viene utilizzata la ricerca in ampiezza (breadth-first search) e una struttura ad albero **Hash** per contare in modo efficiente i set di elementi candidati. Vengono generati insiemi di elementi candidati di lunghezza k da insiemi di elementi di lunghezza $k-1$, quindi sfofoltisce i candidati che hanno un sottomodello poco frequente. L'insieme candidato contiene così tutti gli insiemi di elementi frequenti di lunghezza k . Successivamente, esegue la scansione del database delle transazioni per determinare gli insiemi di elementi frequenti tra i candidati.

Lo pseudo codice per l'algoritmo è riportato di seguito. Viene impiegata la notazione della teoria degli insiemi. Ad ogni passo, si assume che l'algoritmo generi gli insiemi candidati dagli insiemi di elementi del livello precedente.

```
frequentPatternDiscovery (DTarget,minS) →FP
begin
    FP = ∅
    L_1 = {1 item che compaiono in minS x |D| transazioni di DTarget}
    K=2
    while L_{k-1} ≠ ∅ do
        begin
            C_k = {candidati generati da L_{k-1} aggiungendo un nuovo item}
            L_k = ∅
            for each (p ∈ C_k) do
                if (supporto(p, DTarget) ≥ minS) then
                    L_k = L_k ∪ p
            FP = FP ∪ L_k
            K = K + 1
        end
    return FP
end

EPDiscovery(DBackground,FP,minGr)→EP
begin
    EP = ∅
    for each (p ∈ FP) do
        begin
            if (growrate(p,DBackground) ≥ minGR) then
                EP = EP ∪ p
        end
    return EP
end
```

L'algoritmo Apriori soffre di una serie di inefficienze:

- la generazione di candidati genera un numero elevato di sottoinsiemi; infatti, l'algoritmo tenta di caricare il gruppo di candidati con il maggior numero possibile di sottoinsiemi prima di ogni scansione del database;
- l'esplorazione dal basso verso l'alto del sottoinsieme trova qualsiasi sottoinsieme massimale S solo dopo tutti i $2^{|S|} - 1$ suoi sottoinsiemi propri;
- l'algoritmo esegue la scansione del database troppe volte, il che riduce le prestazioni complessive. Per questo motivo, l'algoritmo presuppone che il database sia permanentemente in memoria;
- sia la complessità temporale che spaziale di questo algoritmo sono molto elevate: $O(2^{|D|})$, quindi esponenziale, dove $|D|$ è il numero totale di elementi presenti nel database.

2. Guida all'Installazione

Prima di procedere con l'installazione del progetto, assicurarsi che la cartella "Progetto_MAP" si trovi sul desktop.

2.1 Installazione server

Per la corretta installazione del progetto lato server è necessario:

- MySQL 8.0; (<https://dev.mysql.com/downloads/installer/>)
- Java Runtime Environment (JRE) versione 16; (<https://www.oracle.com/java/technologies/javase/jdk16-archive-downloads.html>)²
- linkare al progetto relativo al server il file "mysql-connector-java-8.0.17", responsabile della connessione al database MySQL.³
Tale connettore è reperibile nella cartella "SQL Connector e script";
- avviare il server MySQL;
- eseguire lo script MySQL presente nella cartella "SQL Connector e script".
Tale script inizializza il database con tabelle e tuple di esempio.⁴

Per un più immediato utilizzo, la porta 3306 è preimpostata per accedere al database MySQL.

2.2 Installazione client

Per la corretta installazione del progetto lato client è necessario:

- installare Java Runtime Environment (JRE) versione 16; (<https://www.oracle.com/java/technologies/javase/jdk16-archive-downloads.html>)³

Per un più immediato utilizzo, l'indirizzo IP al quale il client si connette è preimpostato a quello locale 127.0.0.1 con porta 8080.

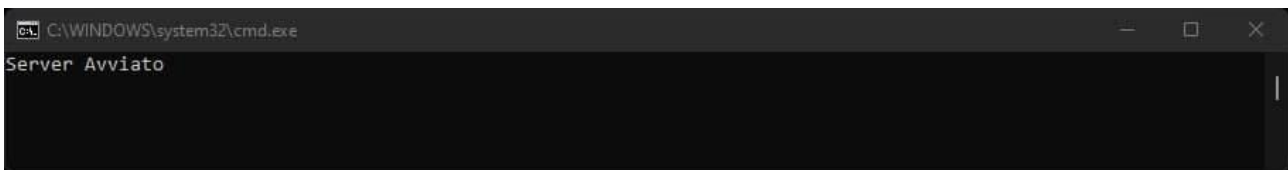
3. Guida utente

3.1 Applicativo server

È possibile avviare il programma nelle seguenti modalità:

1. Esecuzione del file Server.bat nella cartella
“\Progetto_MAP\EpMiner_esteso\Server” (su ambiente Windows).
2. Esecuzione del jar (multiplatform) da terminale:
 - Aprire un nuovo terminale (prompt dei comandi, bash, powershell...);
 - Posizionarsi sulla cartella contenente il file jar,
“<path>\Progetto_MAP\EpMiner_esteso\Server”;
 - Eseguire il comando java per l’apertura dell’applicativo:
java -jar Server.jar

1) Avvio del server:

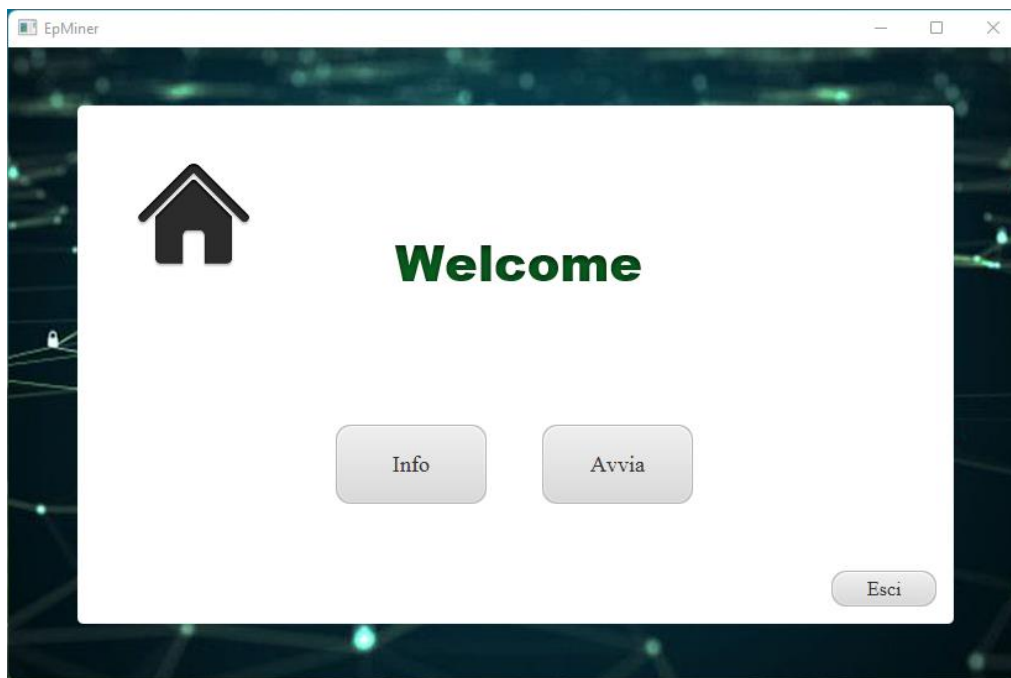


3.2 Applicativo client

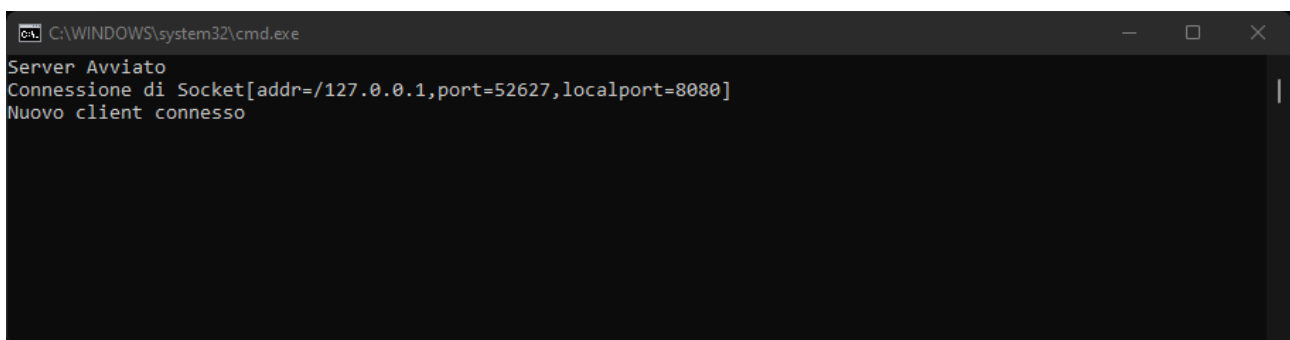
È possibile avviare il programma nella seguente modalità:

1. Esecuzione del jar (multiplatform) da terminale:
 - Aprire un nuovo terminale (prompt dei comandi, bash, powershell...);
 - Eseguire il comando java per l’apertura dell’applicativo:
java --module-path Desktop\Progetto_MAP\EpMiner_esteso\javafx-sdk-17.0.1\lib --add-modules javafx.controls,javafx.fxml -jar Desktop\Progetto_MAP\EpMiner_esteso\Client\Client.jar

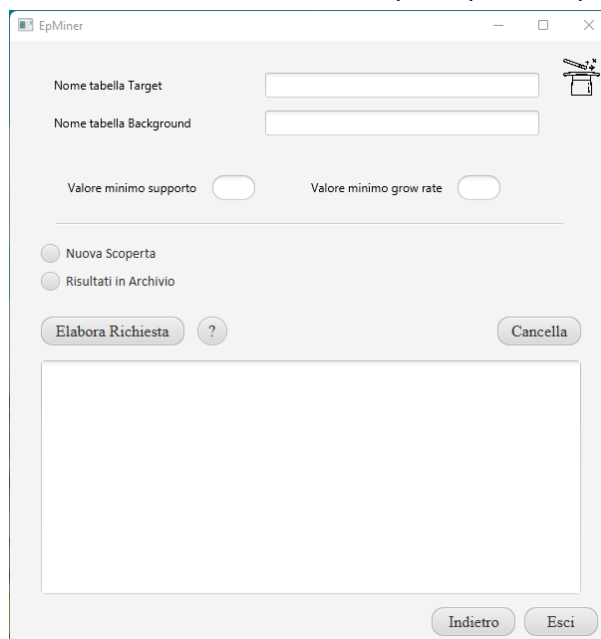
2) Avvio del client:



3) Connessione del client al server:



A connessione avvenuta si visualizzerà la schermata principale del programma:



4) Nuova Scoperta:

EpMiner

Nome tabella Target: playtennistarget

Nome tabella Background: playtennisBackground

Valore minimo supporto: 0.3 Valore minimo grow rate: 1

☒ Nuova Scoperta
☐ Risultati in Archivio

Elabora Richiesta ? Cancella

Frequent Patterns:

- 1:(outlook=rain)[0.375]
- 2:(outlook=sunny)[0.375]
- 3:(temperature in [0.0,6.0721197])[0.375]
- 4:(temperature in [24.288479,30.3606])[0.375]
- 5:(umidity=normal)[0.375]
- 6:(wind=strong)[0.375]
- 7:(outlook=sunny) AND (umidity=high)[0.375]
- 8:(outlook=sunny) AND (play=no)[0.375]
- 9:(temperature in [0.0,6.0721197]) AND (umidity=normal)[0.375]
- 10:(temperature in [24.288479,30.3606]) AND (umidity=high)[0.375]
- 11:(umidity=high) AND (outlook=sunny)[0.375]
- 12:(umidity=high) AND (temperature in [24.288479,30.3606])[0.375]

Indietro Esci

5) Risultati in archivio:

EpMiner

Nome tabella Target: playtennistarget

Nome tabella Background: playtennisbackground

Valore minimo supporto: 0.3 Valore minimo grow rate: 1

☐ Nuova Scoperta

☒ Risultati in Archivio

Elabora Richiesta ? Cancella

Frequent Patterns:

- 1:(outlook=rain)[0.375]
- 2:(outlook=sunny)[0.375]
- 3:(temperature in [0.0,6.0721197])[0.375]
- 4:(temperature in [24.288479,30.3606])[0.375]
- 5:(umidity=normal)[0.375]
- 6:(wind=strong)[0.375]
- 7:(outlook=sunny) AND (umidity=high)[0.375]
- 8:(outlook=sunny) AND (play=no)[0.375]
- 9:(temperature in [0.0,6.0721197]) AND (umidity=normal)[0.375]
- 10:(temperature in [24.288479,30.3606]) AND (umidity=high)[0.375]
- 11:(umidity=high) AND (outlook=sunny)[0.375]
- 12:(umidity=high) AND (temperature in [24.288479,30.3606])[0.375]
- 13:(umidity=high) AND (play=no)[0.375]

Indietro Esci

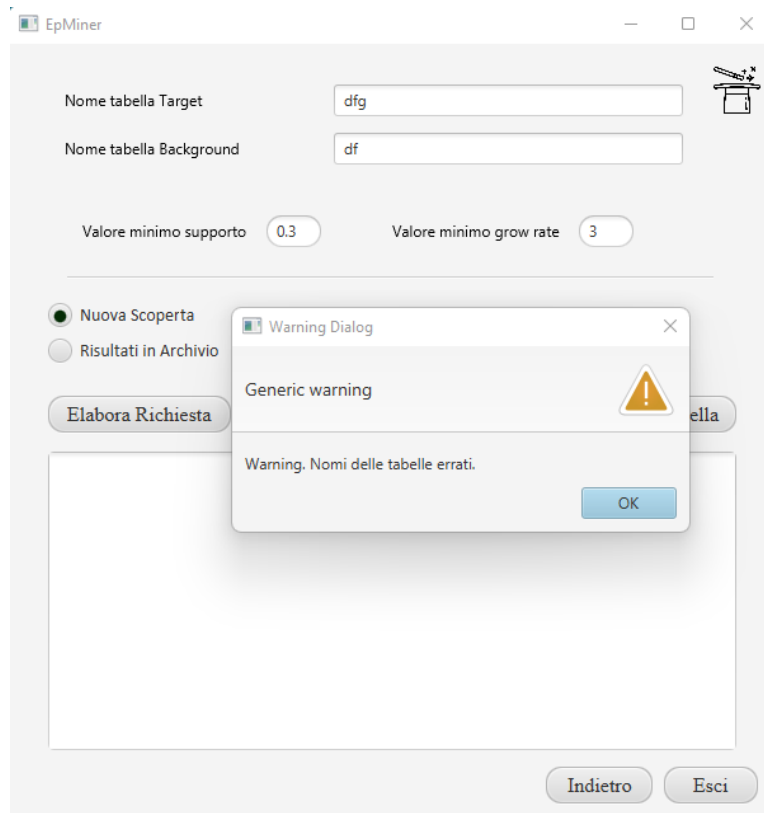
Per chiudere la connessione del client basta premere il tasto “Indietro” e si tornerà alla schermata di avvio.

6) Client disconnesso (lato server):

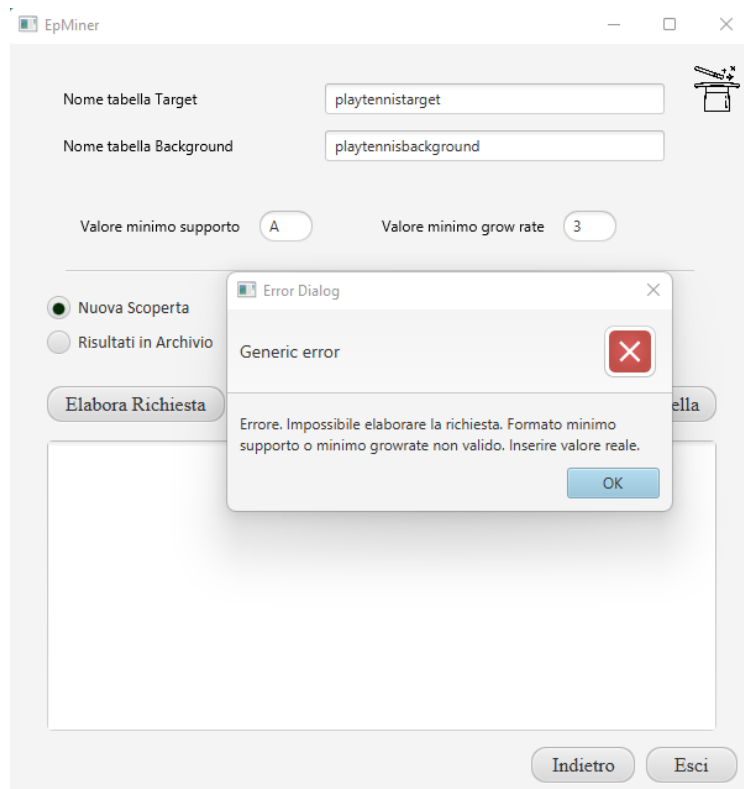
```
C:\WINDOWS\system32\cmd.exe
Server Avviato
Connessione di Socket[addr=/127.0.0.1,port=52627,localport=8080]
Nuovo client connesso
java.net.SocketException: Connection reset
Socket closed
```

Possibili errori:

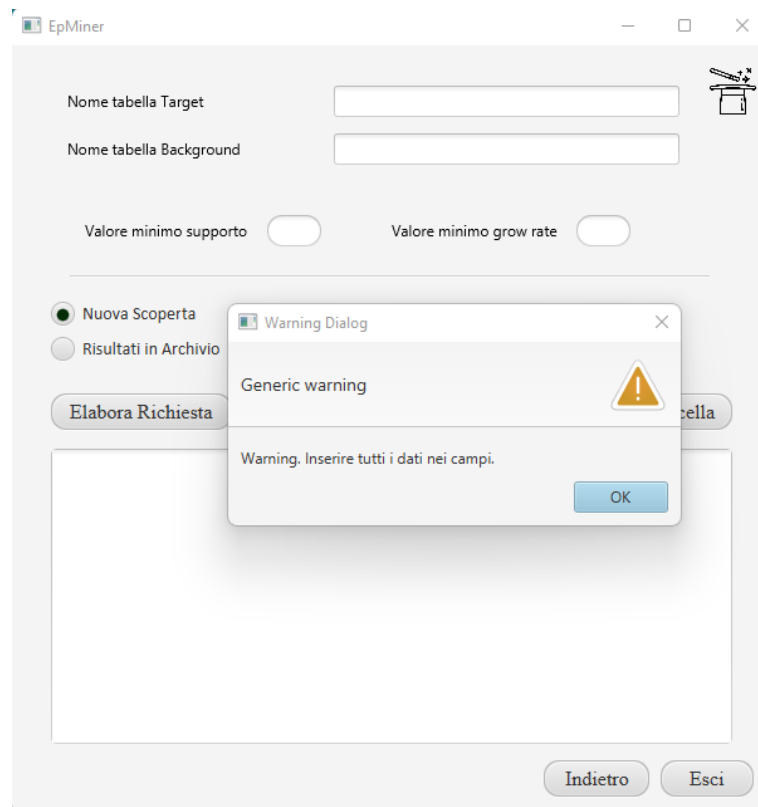
1) Nomi delle tabelle errati:



2) Valore minimo supporto/grow rate errato:



3) Campi vuoti:



Note

¹ Per scelta stilistica non è stata implementata la possibilità di ridimensionare la schermata dell'applicazione grafica.

² Nel caso in cui il file `.bat` o il file `.jar` non venga eseguito correttamente, assicurarsi che l'indirizzo della cartella della JDK sia aggiunta alla variabile d'ambiente `PATH`.

(https://www.java.com/it/download/help/path_it.html)

³ Nel caso in cui il file `jar` non funzionasse a causa del connettore non linkato correttamente, è possibile importare quest'ultimo come libreria esterna al progetto, ed avviare il server direttamente da IDE.

⁴ In alternativa si può aprire il file con un editor di testo e copiare il contenuto nella Shell MySQL.