

Quality Threshold Clustering

Ivan Diliso Matricola: 676366

20-03-2020

Contents

1	Introduzione	2
2	Applicazione principale	2
2.1	Server	2
2.2	Client	2
3	Estensione	2
3.1	Web Server	3
3.2	Libreria JSON	3
3.3	Frontend	3
4	Installazione	4
4.1	Server	4
4.2	Client	4
5	Manuale Utente	4
5.1	Client	4
5.2	Server	4
5.3	Server Estensione	4
5.4	Client Estensione	5
6	Esempi utilizzo	5
6.1	Progetto Base	5
6.2	Estensione	7
7	Note	11
7.1	Importare il progetto in Eclipse	11
7.2	Importare il progetto in IntelliJ Idea	11
7.3	Compilare l'applicazione	11

1 Introduzione

Il software **Qt-Clustering** realizzato in Java permette di applicare l'algoritmo di cluster Quality Threshold ad un insieme di dati estratti da un database MySQL.

Quality Threshold Algoritmo alternativo per partizionare i dati. Richiede piu' potenza di calcolo rispetto a K-Means ma non richiede di specificare il numero di cluster a priori, e restituisce sempre lo stesso risultato quando si ripete diverse volte.

2 Applicazione principale

L'applicazione e' divisa in due componenti principali

2.1 Server

Il server si occupa di eseguire le richieste di uno o piu' client fornendo ad essi le seguenti funzionalita:

- Caricare una tabella dal database specificandone il nome
- Eseguire il clustering dei dati caricati con un raggio specificato dall'utente
- Visualizzare i dati e il risultato del clustering
- Salvare il risultato del clustering su file
- Caricare i dati del clustering da file

2.2 Client

Il client si occupa di gestire una interaccia utente basata su linea di comando che permette all'utente di:

- Connettersi al server specificando una porta (valore di default 8080)
- Chiedere al server di eseguire una nuova operazione di clustering specificando nome della tabella, raggio e nome del file dove salvare i risultati
- Caricare un clustering salvato su file specificando il nome del file

3 Estensione

L'estensione del progetto consiste su una Web Application basata sul framework Spring Boot, che permette di creare applicazioni web basate sul pattern architettonico MVC (Model View Controller). L'applicazione e' basata sull'utilizzo di servlet per servire le pagine web, non sara' piu' presente dunque un modello

client server basato su socket ma il client sara' rappresentato da una pagina web che si interfaccera con il server per l'esecuzione del clustering, il caricamento dei dati dal database e il trasferimento dei dati.

Servlet Oggetto scritto in linguaggio Java che opera all'interno di un server web permettendo quindi la creazione di applicazioni web.

3.1 Web Server

E' stata utilizzata una versione che opera tramite il web server e servlet container Tomcat embedded all'interno dell'applicazione. In questo modo non ci sara' bisogno di avviare il web server Tomcat ma bastera' avviare il WAR (Web Application Resource) del server per avviare sia il web server, sia la web application

3.2 Libreria JSON

Per servire i dati del database e i risultati del clustering sono state create delle classi per la costruzione di file JSON. La libreria presenta una struttura ad albero per rappresentare i dati.

3.3 Frontend

Il frontend e' stato realizzato in HTML/CSS e presenta delle funzioni in Javascript per trasformare i dati JSON forniti dal server in tabelle HTML. E' stata utilizzata la libreria BOOTSTRAP per omologare la grafica delle pagine web e fornire dei siti web responsive

4 Installazione

Questi passaggi sono validi sia per il progetto base sia per il progetto con estensione. Sia il client sia il server necessitano l'installazione dei seguenti software:

- MySQL
- Java Runtime Envrioment 8.0

4.1 Server

1. Accertarsi che il server MySQL sia attivo sulla propria macchina
2. Eseguire il file SQL databasesetup.sql tramite il comando:

```
mysql -u root -p < databasesetup.sql
```

4.2 Client

Il client sia nel progetto base sia nel progetto con estensione non richiedono particolari passaggi per l'installazione

5 Manuale Utente

5.1 Client

Avviare il client facendo doppio click sul file **Client.bat** in questo caso verranno utilizzati indirizzo IP e porta di default (localhost:8080). Per specificare un diverso indirizzo ip e port usare il comando:

```
QtClient.bat <ip> <porta>
```

5.2 Server

Avviare il server facendo doppio click sul file **Server.bat** in questo caso verra' utilizzata la porta di default (8080). Per specificare una diversa porta usare il comando:

```
QtServer.bat <porta>
```

I file creati dal server verranno salvati nella stessa cartella in cui e' contenuto il file bat

5.3 Server Estensione

Avviare il server facendo doppio click sul file **WebServer.bat** in questo caso verra' utilizzata la porta di default (8080). Per specificare una diversa porta

usare il comando:

```
QtWebServer.bat <porta>
```

I file creati dal server verranno salvati nella stessa cartella in cui e' contenuto il file bat

5.4 Client Estensione

Accedere al sito web alla pagina:

```
https://localhost:8080/index
```

6 Esempi utilizzo

6.1 Progetto Base

In tutte le immagini le parole in verde sono input forniti dall'utente

```
[INFO] Connected to sever: localhost:8080

Main menu
[1] Load clusters from file
[2] Calculate clusters from database data

Choose one option (1-2): 1
Filename: mario
mario.dmp (No such file or directory)

Would you choose a new operation from menu?
[1] YES
[2] NO

Choose one option (1-2):
```

Figure 1: L'utente sceglie di caricare i risultati del clustering da file. Prova quindi a caricare un file chiamato "mario" ma il file non e' presente quindi viene visualizzato un errore. All'utente viene chiesto se vuole scegliere una nuova opzione dal menu

```
[INFO] Connected to sever: localhost:8080

Main menu
[1] Load clusters from file
[2] Calculate clusters from database data

Choose one option (1-2): 2
Table name: playtennis
Radius: 2
Number of Clusters:3
1:Centroid=(sunny 30.3 high weak no )
Examples:
[sunny 30.3 high weak no ] dist=0.0
[sunny 30.3 high strong no ] dist=1.0
[sunny 13.0 high weak no ] dist=0.5709571100859968

AvgDistance=0.5236523700286656
```

Figure 2: L’utente sceglie di effettuare il clustering su tabella su database, inserisce quindi il nome della tabella e il raggio. Nell’immagine viene visualizzato solo uno dei tre cluster forniti in output.

```
Filename: mario

Would you repeat?
[1] YES
[2] NO

Choose one option (1-2):
```

Figure 3: Dopo aver visualizzato i risultati viene chiesto all’utente di inserire il nome del file in cui salvare i risultati. L’utente puo’ decidere di effettuare nuovamente la computazione sulla stessa tabella fornendo soltanto il raggio.

```

Main menu
[1] Load clusters from file
[2] Calculate clusters from database data

Choose one option (1-2): 1
Filename: mario
Centroid=(sunny30.3highweakno)
Centroid=(overcast12.5highstrongyes)
Centroid=(rain0.0normalweakyes)

```

Figure 4: Scegliendo di caricare i cluster da file e inserendo un nome di file corretto vengono visualizzati i dati dei cluster.

6.2 Estensione

E' stato utilizzato il browser Firefox per le prove di utilizzo della applicazione web

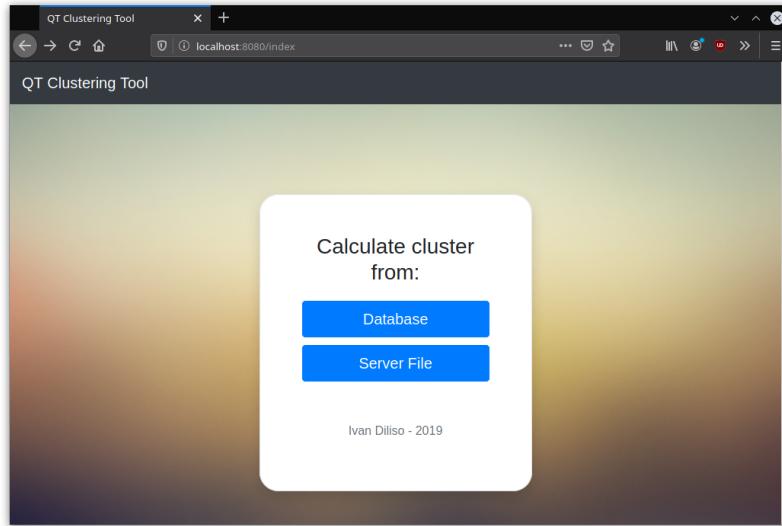


Figure 5: Menu principale, per caricare effettuare il clustering su database cliccare il pulsante "Database", per caricare i risultati del clustering da file cliccare il pulsante "Server File"

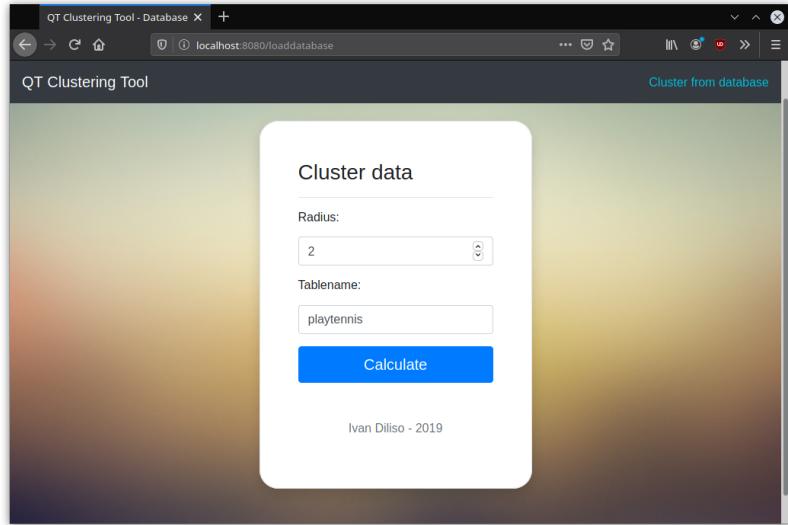


Figure 6: Cliccando il pulsante ”Database” si arriva su questa pagina. L’utente ha inserito un raggio e il nome della tabella da caricare dal database

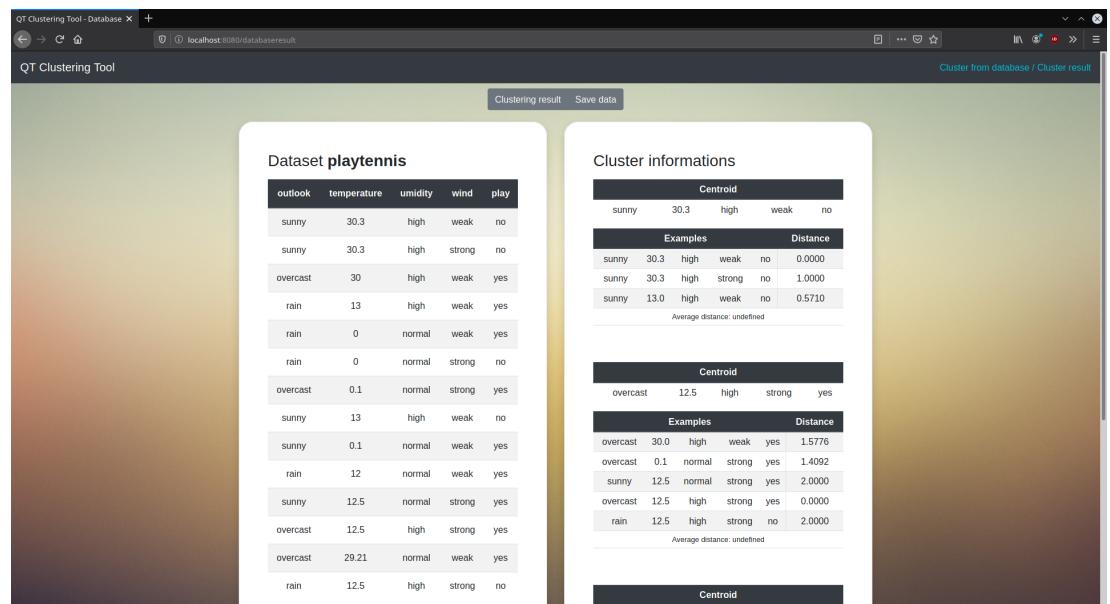


Figure 7: Vengono visualizzati i risultati del clustering. Cliccare su ”Save Data” per salvare i risultati su file

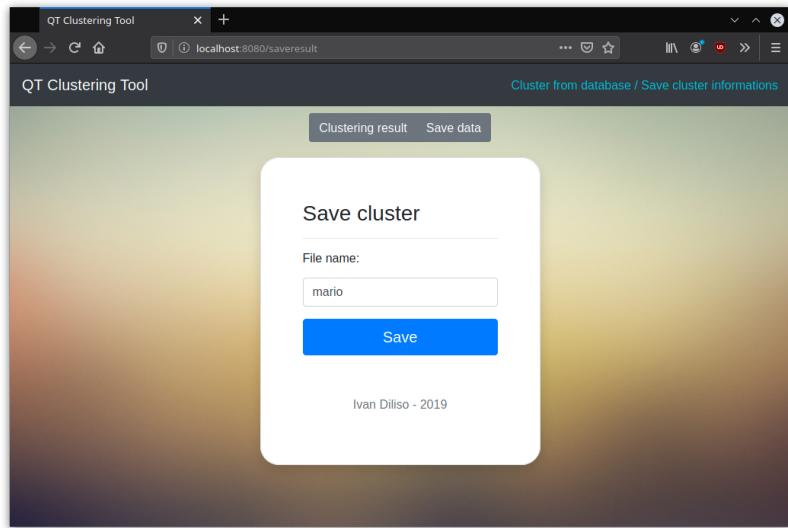


Figure 8: L'utente inserisce il nome del file sul quale salvare i risultati del clustering

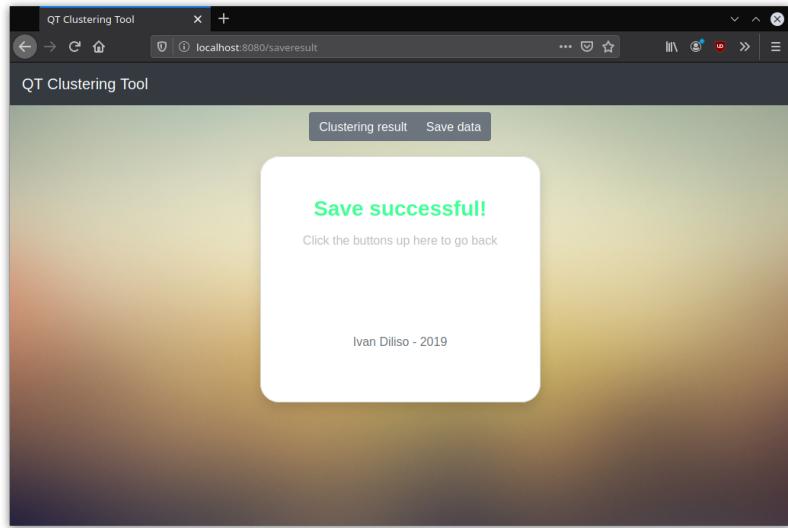


Figure 9: Salvataggio su file avvenuto con successo

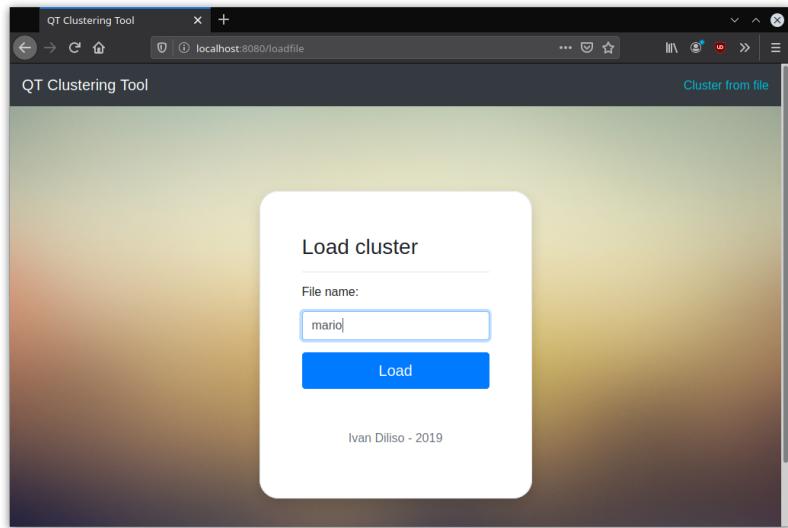


Figure 10: Cliccando su "Server File" nel menu principale si arriva a questa pagina. L'utente inserisce il nome del file dal quale caricare i dati del clustering

A screenshot of a web browser window titled "QT Clustering Tool". The address bar shows "localhost:8080/fileresult". The main content area has a dark background with a central white dialog box. The dialog box is titled "Cluster result : Centroids" and contains a table with the following data:

outlook	temperature	umidity	wind	play
sunny	30.3	high	weak	no
overcast	12.5	high	strong	yes
rain	0	normal	weak	yes

Figure 11: Dati del clustering caricati da file.

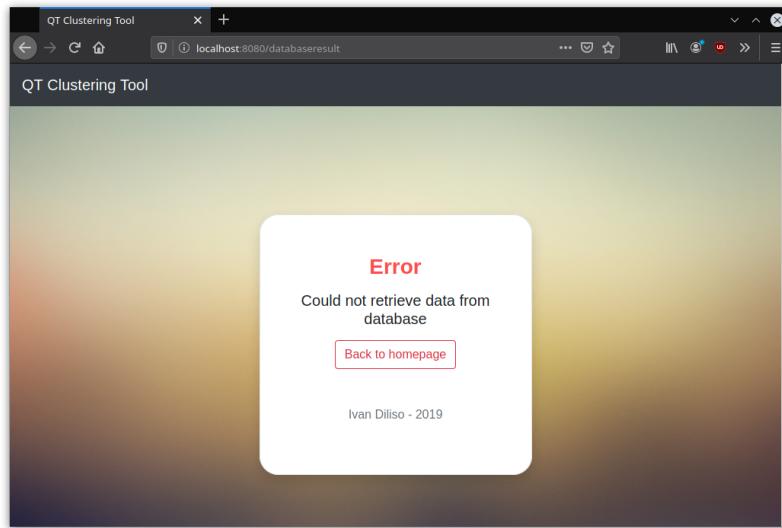


Figure 12: Pagina in cui vengono visualizzati eventuali errori del server.
Cliccare sul pulsante "Back to Homepage" per tornare alla Homepage

7 Note

Nello sviluppo del progetto e' stato utilizzato il software **Git** per il controllo di versione e il software **Gradle** come sistema di build e gestore delle dipendenze

7.1 Importare il progetto in Eclipse

File > Import > Gradle > Existing Gradle Project > Next > Selezionare la cartella QtClient / QtServer / QtWebServer > Next > Finish >

7.2 Importare il progetto in IntelliJ Idea

Import Project > Selezionare la cartella QtClient / QtServer / QtWebServer > Import project from external model > Selezionare Gradle > Finish

7.3 Compilare l'applicazione

Utilizzare il task Gradle "build" per compilare l'applicazione