List of Examples for Final Project in EE5178: DBMS from SQL to NoSQL

Ming-Ling Lo

The examples below are chosen from real student projects in the past, from Academic year 106 to 111

Selected Projects from Academic Year 106

Column-Based Select with MySQL

Abstract

在這個 Machine Learning 盛行的時代，時常需要分析不同來源的資料，然而資料會有不少的雜訊，因此在做 training 前需要選擇一些好的 feature。我們提供了使用者一個方便篩選 feature (attribute)的系統，利用了 python 的 subprocess 連接 MySQL 做出一個加強版的 shell，定義我們的自訂操作 SELECTA，parse 後用 pymysql 來存取資料庫。我們提供了一些常用的 aggregation function，例如 correlation 和 standard deviation，並且在效能上做了一些簡單的分析，提出了一些優化的方案。

Teacher's comment:

Excellent in multiple ways. One of the best projects over these years. Amazed to this project came out of senior year (大三) students.

Joins for Online Aggregation

Abstract

現今資料庫中資料量越來越龐大，若想要進行 online join aggregation，時常會因為需要大量的計算資源及時間而無法給予即時的回應。而由過去的經驗得知，使用者請求 join aggregation 之結果時，多半可以允許一定程度內的誤差。因此，許多研究面對 join aggregation 的問題採取抽樣後計算的做法，期望能在短時間內得到可接受的估計值。本文整理了四篇相關的研究，其中有兩篇分別提出 online join aggregation的方法，一篇討論要如何在分散式系統上提升效率，以及一篇評估取樣效果的優劣。最後我們進行簡單的模擬來比較不同方法間的效能差異。

Teacher's comment:

課堂上並未講解online join, 本組研究online join的result approximtion問題, 分析各papers優劣, 並設計實驗實作, 以數據呈現分析結果.

Paper Pool with Recommendation System

Abstract

Mendeley(圖一)是一個免費書目管理工具與社群媒體平台，能幫助使用者管理與組織研究、與其他同儕合作並同時發現新的研究領域。Mendeley可以自動提取論文中的相關資訊比如title、journal、authors等，並可以用平實的組織管理方式來配合工作流程。

但雖擁有各種強大的功能，我們在實際使用下發現Mendeley仍有許多不方便的地方，例如在網頁版使用中，並沒有提供search的功能，導致使用與管理上有

極大的困難。再舉例來說，當我們在看完一個系列或是好幾篇相關的論文時，會希望做一個小型的群組並給予系列文章一個統整性的評論，有助於平台使用者間的知識流通與分享，但mendeley的架構設計並不支援這樣的結構，綜觀上述缺點，我們希望能夠透過自建database系統並與網頁做連結來達到我們的需求。
Teacher comment:
Combine Mendeley with Relational DB with a recommendation system built by this team

Real-Time Monitoring System
Abstract
We describe a real-time system that is able to perform operations on streaming data to simulate a factory sensor monitoring system. Our system consists of the following parts: (1) a data generator that mimics real-world sensors to output continuous(streaming) data (2) a streaming platform/message hub hosted by Kafka (3) data operation/storage unit implemented by PipelineDB using data streams and continuous views (4) and a webpage for visualization.
We demonstrate that the implemented system is capable of several tasks, including consuming continuous data, performing complex queries and operations on the data streams, and presenting the results on the webpage, all in real time.
Teacher's comment:
Integration of a data generating simulator, a message queue system (Kafka), a time series database system (PipelineDB) and a data visualization system, to implement a factory data monitor system. This team demonstrated their understanding and grasp of systems, and used their knowledge to build a system that was essentially a small version of real-time data monitoring system that can be used in factories.

NewSQL System Study - A Case Study on TiDB
Abstract
NewSQL 是指一種新類型的關聯式資料庫管理系統，它針對 OLTP 實現讀寫工作負載，追求提供和 NoSQL 系統相同的擴展性，且仍然保持傳統資料庫支持的 ACID 特性。在此次系統研讀中，我們研究了 NewSQL 的相關論文與特性，並以一開源的實例：TiDB 來進行更深入的系統探究，最終佐以業務層實作、具規模的部署和效能評測三個方向，來驗證所學的研讀。
Teacher's comment:
NewSQL 是 NoSQL之後更新的DB趨勢之一. 本組以課堂所學出發, 做延伸研究, 並自行設計實驗, 架設開源軟體來研究及驗證他們的觀點. 過程中需解決許多系統問題. 且為驗證自己的觀點, 並設計出一組stateless API所為工具.

Selected Projects from Academic Year 107

A SIMPLE USER DEFINED FUNCTION FOR DATA PLOTTING
Abstract
When dealing with data, most people tend to rely on plots to see the trend or general distribution of data points. Nevertheless, in MySQL, one of the most important and most used database management system in the world, there is no such function to

draw a plot directly from the database. In this project, we create a simple userdefined function to do simple plotting, which executes a python script to achieve this goal.

Teache'rs comment:
Interesting idea. Such functionality can be quite useful. However, the work can be developed deeper.

A minimal DBMS implemented in Rust

Abstract

Database management systems (DBMS) are used in everywhere. There are many DBMSs, including MySQL, MongoDB, etc. We implement a minimal DBMS supporting SQL in Rust in a new project, StellarSQL (https://github.com/tigercosmos/StellarSQL) from scratch.

Teacher's comment:
Highest score in academic year 107

Graph Database Comparison & Exemplar Query

Abstract

Graph databases have become a popular type of NoSQL due to the fact that they are well-suited to mine data from social media, biological networks and business relations. In this work, we probed some graph databases and built a python query extension on top of Neo4j, using a new query paradigm - Exemplar Query. In the first section, two popular graph databases, including Neo4j, OrientDB are compared in terms of its speed and functionality. In the second part, the algorithm of our exemplar query is first introduced and the effectiveness is evaluated using a minimal data set and a real data set, respectively. The results show its potential for information searching, particularly suitable for the one has as only an element from the desired result.

Selected Projects from Academic Year 108

MySQL-kNN: A MySQL Store Engine supporting K Nearest Neighbor

Abstract

The k-Nearest-Neighbors (kNN) is a simple but effective method for classification. The major drawbacks with respect to kNN is its low efficiency on existing databases. In this paper, we propose a novel DBMS store engine, namely MySQL-kNN, which is built on MySQL for kNN applications that is aimed at overcoming this shortcoming. Our method constructs a store engine using kd-tree, ball-tree for the indexing, which classifies data into its region. The construction of the model reduces the number of data we need to check, and makes classification faster. Experiments were carried out on some datasets generated from the random number generator in order to test our method. The experimental results show that with one million records, we can be about 50 times faster.

Teacher's comment:
This group augmented useful functionality to SQL engine. The system implementation is solid, and the work is complete with experiments to verify the result.
One of two projects with the highest score in year 108.

MengoDB, High performance remote database

Abstract:
As one of the most popular databases, MongoDB is favorable for its simplicity of operation for users. Users' high usability is still inadequate to belie the defects. The performance of MongoDB is noted when comparing with other SQL databases, but according to the research, MongoDB shows its low performance if we refer to the multi-thread cases, which is often a real-world implementation. In this work, we present a method to cache the data from the MongoDB and record data into Redis, which suggests better performance on data streaming. Background process and optimized policy on pre-fetching is used to ensure the validity and improve the effectiveness. Experimental results demonstrate that Mengo, MongoDB with the aid of Redis can reach higher performance in reading and writing data. Meanwhile, this paper solves the sync-problem and the best cache policy to improve efficiency.
Teacher's comment:
Good and useful idea, solving real-life problem, solid system implementation, complete with implementation performance measurement.
One of two projects with the highest score in year 108.

Map Information Visualization – Graph Database Connect to Google Map
Abstract
我們希望提供一個將Graph Database在地圖App上視覺化呈現的工具，並將不同的新聞透過地區關聯在一起。透過將新聞的資訊導入Graph Database，以及現有的地圖工具API，我們可以將新聞資訊地圖工具做結合，當使用者在地圖上點擊一個地區時，會跳出近期該地區發生的新聞；而藉由query，我們可以找出那些國家或地方之間有重要的互動，例如:簽訂協議。提供一個更方便了解世界局勢的管道。而此將Graph Database資訊在地圖上視覺化呈現的工具，將來也可泛用在不同資料上並供他人使用。
Teacher's comment:
有趣也有用的命題. 整合Graph database 以及Google map API, 展現了對系統的了解. 不過整個work最後沒有做得很完整, 屬於idea優於completed work的一個例子.

Selected Projects from Academic Year 109

整合統計功能與 **SQL -** 使用 **MySQL** 為底層 **SQL** 引擎
Abstract
MySQL 是一個強大的資料庫,裡面有很豐富的功能滿足使用者的需要,但我們發現 MySQL 內建的數學函式相對基本,因此我們希望可以實作一些統計工具,例如:相關係數、取樣、找出資料缺值等等,以及資料視覺化的功能以幫助使用者可以用簡單的操作去完成他們的目標。
我們實作的架構主要是在原生 SQL 引擎上加一個中間層,使用者的指令經過處理後,會透過這個中間層傳給原生的 SQL 引擎。SQL 引擎回傳資料後,我們看使用者用了哪些新增的功能,再將所選的功能套用在資料上,最後回傳最終的結果。

Hang out with Visual Data: Visual Oriented Database
Abstract
Generally, most database systems only support simple storage and reading of the image data and lack image analysis functions. Thus, we opt to create a visual-oriented database that can make up for the shortcomings of the current database in processing image data. We implement a database with a socket-based interface aimed at image

storage, and query functionality, which exploits image process, compression, and machine learning techniques.

Teacher's comment:

Highest score in year 109

MySQL Data Analysis

Abstract

With today's data-driven organizations and the introduction of big data, data analytics are often overwhelmed with the amount of data that is collected. Taking the time to pull information from remote server-side databases and put it into another analysis tool is frustrating and time-consuming. Therefore, we decided to improve the database functionality through data analysis function implementation on MySQL server.

There is a need for a database management system that directly supports basic data analysis functions without data transmission and collection process. Our implementation on MySQL server includes various statistical functions and data fitting functions. Employees and decision-makers will have access to the real-time information they need in an appealing format.

Teacher's comment:

To add statistical or even ML capabilities into DBMS is the correct trend in the industry. Good topic.

TimeSeries ARIMA for Redis

Abstract

Abstract—Time series analysis can be useful to see how a given asset, security, or economic variable changes overtime. Redis, as a NoSQL key value database, is suitable to save the time series data and in-memory designed style make it be queried more efficiently. When combining the traditional ARIMA model on the Server side, the client side will be able to forecast Time series data in a more concise way.

Teacher's comment:

Implementing ARIMA time series analysis capability into the timeSeries module of Redis

InvaliDB Implementation

Abstract

In this generation network has become indispensable, shopping or performing bank operations online is fairly normal in our daily life. However, the rise of these kinds of web services demands nowadays databases or servers to detect and publish changes with low latency. Given that traditional databases are mostly pull-based, which would output data only when receiving queries. Since they are excelling in dealing with slowly-changing large amounts of data, not rapidly-changing small amounts of data. Therefore, real-time databases that can handle high frequency data change and publish changed data in a short time are getting more popular these days. Some real-time databases such as MeteorDB, RethinkDB and Firebase have received great reputation and are widely-used today. However, the query mechanisms of these databases are encountered with the challenge of read and write scalability and a lot of research has been done managing to fix this bottleneck.

Seeing the problem listed above, we focus on soft real-time databases which accept a deadline violation, and try to implement a real-time database that can support push-based queries based on a pull-based database, called InvaliDB. We claim that it is a solution to the scalability problem. Our implementation is a software solution instead of the hardware method our referenced paper proposed. In addition, we design our experiments, profiling the scalability of software InvaliDB.

Teacher's comment:

Implementing support for push-based query on InvaliDB.

Selected Projects from Academic Year 109

Implementation of Clustering Toolkit in MySQL

Abstract (生機系)

機器學習為現今主流之資料分析方法,資料科學家透過 Python 與其相關之機器學習框架 package 將資料從資料庫提取出後進一步分析資訊。然而近些年需要分析的資料往往都非常大量,將資料從資料庫存出來再做分析會變得費時且耗資源,所以本團隊希望能在資料庫上實現一些初步的分析, 而分群演算法可以了解資料集中有哪幾群資料彼此是較為相似的。過去 SQL 使用者若要直接在資料庫內實現資料分組需透過人工交互比對的方式選取分組依據以及每組的範圍,這在資料量大且資料特徵很多的時候是沒效率的。而分群演算法可以幫助使用者解決這個困擾。因此,本團隊在資料庫管理系統 MySQL 建立 clustering 的函式,實現直接在資料庫系

統用機器學習方法分析資料,以省去資料傳輸時間,並提供SQL 使用者使用其熟悉之程式語言執行機器學習分析。程式碼可於 https://github.com/Dawson-ma/MySQL-clustering 查看。

Teacher's comment:

The topic makes sense, and can save people's time, particularly during the data exploration phase. This team also finished their work to a certain degree, and put their code on GitHub. (本組在第一階段成績僅中上, 但在final write report 時做得很好.)

JOURS: JOIN and UNION operations recommendation system

Abstract

Schema matching finds similar columns from distinct sources, and it is widely used in data cleaning and integration. Generally, tables can be combined when they share the same information. In the database management system, JOIN and UNION are operations for combining tables. In this study, we developed JOURS, a JOIN and UNION operations recommendation system. With JOURS, similar columns from two tables can be identified, and the similarity score of two columns is calculated. JOURS supports both string and numeric data type columns and compares columns based on their column name. When users import two tables, a recommended table would be generated by JOURS. The recommended table contains the details of columns and the corresponding similarity scores, and it can be a reference when combining tables.

Teacher's comment:

Schema matching is an important and very relevant problem. This team implemented a reasonable solution suitable as a semester project. They finish their implementation and were able to produce demos successfully.

Deep Entity Matching System with Self-supervised Pre-Trained Language model

Abstract
Entity Matching (EM) refers to the problem of determining whether two data entries refer to the same real-world entity. The objective is to determine the set of pairs of data entries, one entry from each table so that each pair of entries refer to the same real world object.

To this end, we propose a Deep Entity Matching System with Self-supervised Pre-trained Language model, which is a complete EM system combining DITTO with self-supervised learning into MySQL database management system (DBMS). Attribute to self-supervised learning (SSL) we adapted, our system is forced to learn “harder” to improve the model’s matching capability. Some dark knowledge and discriminative representations are also acquired from the learning process. Comprehensive experiments on different real-world large-scale EM benchmarks clearly demonstrate the superiority of our approach. Finally, we also cleverly integrate Blocker, Matcher and DBMS these three separate components together and provide users a convenient end-to-end EM solution.
Teacher’s comment:
Entity matching is also a very practical problem to solve. This team combines Ditto, an existing solution with a self-supervised learning approach to improve the system. They also completed their implementation.

Large-scale object detection dataset retrieval
A MySQL Database made for Large-scale Image Retrieval
Abstract:
Data retrieval has been an important task for decades. Recently, the rapid improvement in deep learning is making data retrieval more important since it takes a huge amount of data to train a deep neural network. Furthermore, image-related tasks are one of the most popular topics in this area. Therefore, whether we retrieve image data fast enough has become a crucial task. In this report, we target cropped images retrieval from object detection dataset. Intuitively, all images and bounding box information would be loaded into client-side memory to perform cropping later on. However, it would lead to both memory issues and excessive data transmission. Therefore, we propose a system architecture to overcome this problem by utilizing database properties by processing all of the data on the server side. Additionally, we make a “reverse image search” function by implementing image similarity matching in MySQL. Finally, we propose a more complete framework that incorporates popular object detection models into our system.
Teacher’s comment:
This team tried to augment MySQL with the capability to perform image retrieval, both based on the name of bounding boxes, and based on image similarity. The idea is very great! However, the development of their ideas and the implementation was not very complete. This team received a medium-high score.

StaDB: Analyzing Your Research with Statistics Database
Abstract:
SQL 作為第三大受歡迎的程式語言,其強大的資料管理能力可以促進資料導向(data-driven)的人工智慧(artificial intelligence)的發展。我們以自身的專業出發,開發一套,含有機器學習演算法的資料庫系統,StaDB,其根基建立在一開源套件 SQLflow。在本系統中,增加 Sklearn、PyTorch 等

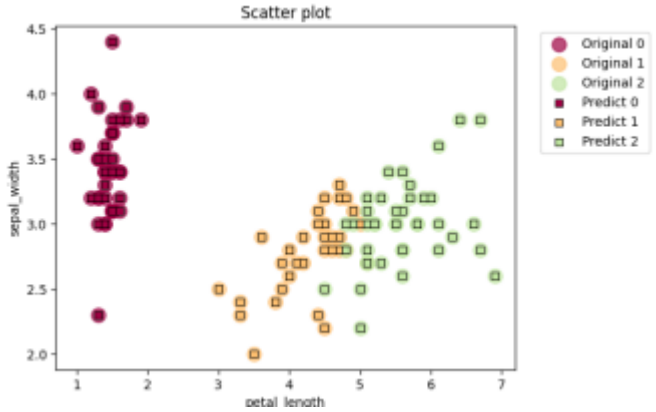
人工智慧套件,並且以擴展語法,增加如:TO TRAIN、TO EVALUATE 等關鍵字,供開發人員使用。此外,系統架構以分散式的形式建立,各個組件可彈性調整,如:AI 引擎可自行擴增其他常用套件。系統中,可調度函式、模型如:分類(classification)、回 歸(regression)、分 群(clustering)、降 維(dimension reduction)等 常 用 機 器 學 習 演 算 法 , 也支援以 PyTorch 寫成的數值資料分類及回歸。模型所輸出的結果,可以透過 Shap、Matplotlib 等分析繪圖套件視覺化。最後,本系統以常見的使用者介面形式,建立一 Web APP 供使用者即時使用。


```
SELECT * FROM iris.train
TO TRAIN Sklearn.SVC WITH
  model.kernel="rbf"
COLUMN
  petal_length,petal_width …
LABEL
  class
INTO my_models.svc;
```

```
SELECT * FROM iris.test
TO EVALUATE my_models.svc
LABEL
  class
INTO iris.evaluate;
```

```
SELECT * FROM iris.test
TO PREDICT iris.predict
USING my_models.svc;
```

```
SELECT * FROM iris.train
TO EXPLAIN my_models.svc
WITH
  summary.choose_features=
  "petal_length,sepal_width'
USING
    Scatter_plot;
```



Teacher's comment:

The topic makes a lot of sense. It can save people a lot of time, when people want to do data exploration using light-weight deep learning approaches. They also did a very good job in the system's work, not only modified and augmented SQL to implement new functions, but also finished their implementation.