Analysis, Matching, and Prediction of Data with Time or Space Characteristics according to Association Rule of Data Mining

*Progress Report*

# Project Summary

|  |  |  |
| --- | --- | --- |
| Report Date | Project Name | Prepared By |
| July 5, 2017 | Analysis, Matching, and Prediction of Data with Time or Space Characteristics according to Association Rule of Data Mining | Shao Jialin, Zhang Pengyu, Jiang Minyu, Zhang Hongbin, Wang Yu\* |

# Status Summary

## Introduction

Low efficiency in massive data query is the biggest problem we need to solve, especially dealing with the data with time and space characteristics. Following steps are what we need to do:

* Learn Golang as the basic method to build up a distributed database.
* Learn several distributed algorithm for building database and data query
* Select data collection and unpacking methods in zip and unzip process.
* Optimize data query algorithm.
* Create different index in storage.

## Project Completion

* **Golang in building database**

The core of distributed database is that original centralized data in database is stored into a plurality of data storage nodes connected by a network, in order to obtain greater storage capacity and higher concurrency amount of access. Thus Golang is more suitable in building distributed database.

Goroutines in Golang is to multiplex independently executing functions into a set of threads. This means when a coroutine blocks, the running program will automatically move the rest of coroutines into other available thread to make sure it would be completely ruined. It only costs a little memory in stack, which takes a few kb except long-time running.

* **Data Collection Method**

With the difficulty in getting a lot of data in personal ability, we need other data in the database as a support. The solution is to send an HTTP request to the API address of the service to get the data we need. Take traffic situation data as an example, access pattern is as follow:

Step 1: Apply for "Web service API interface" key (Key);

Step 2: Stitching the HTTP request URL, which requires to send the key applied form the first step as a required parameter;

Step 3: Receive the return data from HTTP request (JSON or XML format), then parse the data.

If there is no special statement, the input parameters and output data encoding of interface should be uniformly set as UTF-8.

The rectangular area traffic situation query service address is http://restapi.amap.com/v3/traffic/status/rectangle?parameters.

After adding parameter after this API address, we could get the needed traffic information in specific points or areas like this http://restapi.amap.com/v3/traffic/status/rectangle?rectangle=116.351147,39.966309;116.357134,39.968727&key=<用户的key>.

The acquired data is presented in json format, see it in addendum.

* **Data Unpacking Method**

We will get Pre-Data as a data entity after the system pre-processing in acquired data packets, which includes protocol header, protocol content and user’s ID. Pre-Data is a protocol data format which defines a standard, which contains protocol keywords, protocol contents, and contents of user’s ID.

When we need to use some acquired data, we need to unpack the data in json format then pick useful sections up. After unpacking, we can get needed data through extracting keywords among data.

See the code in iOS application using swift language in addendum.

Through this data unpacking method, we could get the data of keyword “username” in server address from “http://localhost:8080/MyServe/userdata.json”

* **Optimization in Data Qeury**

1. **Locally Weighted Regression** **Algorithm**

Locally weighted regression algorithm is the extension of linear regression. Linear regression fits well when the target data is the linear model but it could make big errors in dealing with non-linear data such as the trajectory of “ofo” sharing bike. As a solution, we choose some closed points instead of all points to do linear regression in predicting a point.

Target function: fit to minimize in

When

1. **Logistic Regression Algorithm**

For continuous variable problems, linear regression and locally weighted regression algorithm can solve it. But logistic regression algorithm shows better capability in discrete variable classification problems.

Take an example of "ofo", we could use this algorithm to predict some problem that only have 0 or 1 outcome.

Choose to simulate the objective function: ;

Based on , we can find the classification probability value:

.

Substitute into , we finally get a new update rule of :

1. **Newton's Method Algorithm**

Newton's method algorithm is to search the solution space.

The update rule is: .

* **Contribution of each member**

Data Collection & Database Building: Zhang Pengyu, Jiang Minyu

Machine Learning in Data Query: Shao Jialin, Zhang Hongbin, Wang Yu

## Diversions from plans

**Encountered problems in optimization of data query:**

1. Locally weighted regression algorithm is more suitable for the data set with few data. So when the data set become bigger and bigger, this algorithm's efficiency will flop quickly.
2. In logistic regression algorithm, choosing an appropriate is very essential. Thus we must find more to test the solution.
3. Newton's method algorithm's speed is good enough for our project, but its f(x) has many restricted conditions. So the data set we can use is limited.

## Planning for the coming period

1. Plan schedule



*Figure 1 Updated Plan Schedule*

1. Anticipated potential problems.

Complex testing process of data query algorithms.

Figure out methods to handle the time and space database in correct order.

Ensure the zip and unzip process of real time data could run correctly.

Select the proper protocol solving Internet I/O problems.

# Project Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | % Done | Due Date | Driver | Notes |
| Learning | 80 | 2017/6/9 | All team members | Learning ceph, distributed processing, parallel computing and Golang.  \*Still got a lot of new things to learn along with the promotion of process. |
| Collecting data | 95 | 2017/6/20 | Zhang Pengyu  Jiang Minyu | Data from Ali Yun |
| Optimizing Data query | 100 | 2017/6/20 | Zhang Hongbin  Shao Jialin, Wang Yu | Filtering data. |
| Choosing an available object | 100 | 2017/6/20 | All team members |  |

# Budget Overview

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Category | Spent | % of Total | On Track? | Notes |
| Books | 900 | 90 | Yes | Computer System, Compilers: Principles; Golang; Operating System… |
| Data | 500 | 50 | Yes | \*Cost of data will gain through the increasing data query in the project. |

# Addendum

* **Acquired Data in josn Format**
* **{** "status" : **"1"**,

"info" : **"OK"**,

"infocode" : **"10000"**,

"trafficinfo" :**{**

"description" : **"北三环路：安华桥附近自西向东行驶缓慢。"**,

"evaluation" : **{**

"expedite" : **"84.62%"**,

"congested" : **"15.38%"**,

"blocked" : **"0.00%"**,

"unknown" : **"0.00%"**,

"status" : **"2"**,

"description" : **"轻度拥堵"** **}**

**}**

* **}**
* **Data Unpacking Method in iOS application using swift:**



*Figure 2 Code of Data Unpacking Method in iOS using swift*