PRINCIPAL COMPONENT ANALYSIS ON MySQL DATA RECORDS USING C++

Running our Client Application to Process MySQL data for feature reducion.

PRANJAL MITTAL
ABINASH PANDA

REQUIREMENTS

1)
MySQL (Version 5 or more)

2)
GNU Scientific Library
http://www.gnu.org/software/gsl/

3)
To access MySQL data we first need to install the MySQL Connector for C++.

Installation is possible via Synaptic Package manager on Linux:
libmysqlcppconn5
libmysqcppcon-dev

4) (Optional)

MySQL C++ library bindings (runtime)

MySQL++ is a complex C++ API for MySQL (and other SQL databases soon). The goal of this API is to make working with Queries as easy as working with other STL Containers.

This package provides runtime support.

5) (Optional: But Recommended)
Install **phpmyadmin** to view MySQL data easily on the client side.

Step by Step Instructions to Compile and Run the Code

Step 1:
Change Directory into the code directory.

Step 2:
Run the following command in the terminal to compile the code.
g++ mysqlml.cpp -lmysqlcppconn -lgsl -lgslcblas

Step 3:
Create a database titled "**senocad_ml**" in MySQL usign PhpMyAdmin or console and select it.

Step 4:
Import the file: **"to_be_imported.sql"** via phpmyadmin or LOAD command in MySQL into the **senocad_ml** database.

Step 5:
Set the following parameters defined in Line 26-29 in mysqlml.cpp

#define DB_HOST "**localhost**"
#define DB_USER "**root**"
#define DB_PASS "<password_for_root>"
#define DB_NAME "**mydb_ml**"

Step 6:
Type **./a.out** on the terminal to get the result processed by PCA and printing the result matrix to STDOUT (which can be alternatively inserted into a file or populated into a database as required)

Running Principal Component Analysis Algorithms implemented in C++

Here is a look at our function prototypes using in the PCA header file.

```
#include<gsl/gsl_matrix.h>
#include<gsl/gsl_blas.h>
#include<gsl/gsl_linalg.h>

// Returns the index k at which sum from 0 to k exeeds val.
int sum_till_lessthan(gsl_vector *S,int length, double val)

//function for normalizing the array (x <- (x-mean))
gsl_matrix *normalize(gsl_matrix *mat, int m, int n)

//function for computing the covariance of the array
gsl_matrix *covariance(gsl_matrix *mat, int m, int n)

//function performing PCA on the given matrix; returns the reduced matrix
gsl_matrix *pca(gsl_matrix *mat, int m, int n,int *k)
```

Accessing MySQL Data using MySQL Connector in Cpp

```
// Creates a MySQL driver object to interact with the database
  sql::Driver * driver = get_driver_instance();

// Creates a Connection object to the MySQL database System
  std::auto_ptr< sql::Connection > con(driver->connect(url, user, pass));

// Specifying the database to us.
  con->setSchema(database);

// Preparing an object which will hold the MySQL query statement
  sql::Statement *stmt;

// Creating the Result set object which holds the query results.
  sql::ResultSet  *res;

// Associating the statement with the connectiion object.
  stmt = con->createStatement();

// Quering the database
  res = stmt->executeQuery("SELECT * FROM pca;");
```

The results will not be loaded into the object “res” which can be accesses record by record using res->next()