GGobi and Database Management Systems from GGobi
Reading data from MySQL

September 2, 2000

Abstract

A common source of data is Database Management Systems (DBMS), and specifically relational databases. We describe the facilities currently provided by Ggobi for reading data from MySQL (`http://www.mysql.org`), a particular DBMS.

The ability to read GGobi input from a single XML opens a large number of possibilities. Reading data from a database however is also a powerful extension to any system. Databases offer many conveniences for storing values in comparison with files, URLs, etc. These advantages include

- Large quantities of data can be managed by software designed for that purpose.
- Only a single instance of the data is needed, as applications can dynamically query the segment of that data they are interested *when they need it.* These client applications can be written in any of numerous languages that provide a library or bindings for the DBMS. Support for many of these systems is provided in Python, Perl, Java, C, etc.
- Data can be updated dynamically and made available to clients.
- Updates to the data (e.g. correction of errors) need be performed in a single location, simplifying administration and providing greater data integrity.
- Many of the common, basic computations can be performed using SQL specified in a client process and performed locally on the data within the DBMS.
- Most DBMS provide a security model for limiting access to the data, allowing great resolution for the administrators.

1 MySQL

MySQL is a database management system ...

2 GGobi & MySQL

When one starts GGobi, one can instruct it to use a connection to a MySQL database using the `--mysql`.

```
ggobi -mysql
```

No value need be specified with this flag. However, if a filename is given, the values for establishing the database connection and performing the query are read from that file. The file is a property file consisting of `name: value` pairs. The entries have the same names as those values presented in the GUI described below. See **??**. Any white-space is trimmed away from the values. An example of such a file is given in figure **??**.

This approach allows one to store symbolic information about how to obtain the data. For basic security reasons, the password used to access the dataset should not be stored in this file. In the case that it is, it is read and used directly

```
Host:         tuolomne
User:         duncan
Database:     ggobi
Data query:   SELECT * from flea;
#Password:
Color query: Hi there
```

Figure 1: Default SQL Connections

to establish a connection. If it is omitted from the file, a GUI dialog (described below) is displayed that allows one to interactively set all the values governing the SQL connection. The values from the fields specified in the properties file will be inserted as defaults in the GUI. The user need only fill in the password to complete the connection. In the rare circumstances that no password is needed (e.g. for convenient public access to the database), simply providing an empty value for the Password property is adequate to indicate that a connection should be attempted with no password.

In the case that the `--mysql` is specified with no filename a dialog window is displayed that allows one to interactively specify any and all of the details about the database and the data that one wants. The fields are described in table **??**

host	the name or IP address of the host machine on which the MySQL server (daemon) is running.
user	the login name to use when creating the connection with the database. This is not a login for the machine, but f
password	the password required for the specified user on that database server. This is also set by the administrator when g
database	Identifies which database (collection of tables) within the database server you wish to access.
port	Identifies the port on the host machine on which the database server is receiving connections. It is unlikely that
socket	?
flags	?
Data query	the SQL query that is sent to the server which generates a table (result set) which is treated as the raw data.
Segments query	an SQL query determining which observations are connect. The result should be a table with 2 columns and th
Color query	this is an SQL query that returns a table of RGB values. A fourth column, if present, should contain the names

Table 1: SQL Parameters

The non-data queries are not implemented yet. It would be useful to integrate the data sources so that appearance was specified locally, but the actual data values were queried via SQL.