Recycle of the Netlib/MDS library on the Web

Tatsuki Inoue*

Abstract

A CGI system coded in the Perl language, for calculating multidimensional scaling by using Netlib/MDS, is presented as an example of usefulness of Free Fortran library on the Web. And our system give us a new way to access the data which is not open to the public on the Internet.
Key Words: Information disclosure; Common Gateway Interface(CGI).

1 Introduction

With the development of the Internet, the number of techniques on the web is increasing. CGI is a fundamental technique, which has existed since the first stage of the Internet. It receives parameter from a user via web. So we can construct web page interactively by using this technique. CGI program is started by web-server daemon, e.g. the Apache httpd, on a server, amd it can be received parameter as environment variables from the daemon. Therefore, a stand-alone executable code on the server is available for the CGI system.

CGI program can also call several external procedures at a running time. The external procedure is also a executable code on the server. The difference between the external procedure and CGI program is a process caller. The external command is started by CGI and CGI program is started by web server daemon. The external command has to have three conditions: (1) It is terminated. (2) It can be controlled motion completely at a starting time by parameters via files or via directly. (3) Its output is written by itself on the server.

In this paper, we propose an example of CGI programs which is used Netlib/MDS library as an external program. Netlib[1] is a repository which contains freely available software, documents, and databases of interest to the numerical, scientific computing. It is maintained by AT&T Bell Laboratories, the University of Tennessee and Oak Ridge National Laboratory, and by colleagues world-wide. The collection is replicated at several sites around the world, automatically synchronized, to provide reliable and network efficient service to the global community. Netlib/MDS is also a collection of programs having to do with multidimensional scaling and related methods, including PREFMAP, SINDSCAL (INDSDCAL), ADDTREE, EXTREE, KYST, MDSCAL, HICLUST, and MDPREF (some in multiple versions). These softwares are written in the Fortran language. The Netlib/MDS softwares satisfy

*College of Social Relations, Rikkyo University, Tokyo 171-8501, Japan, tatsu@sr.rikkyo.ac.jp

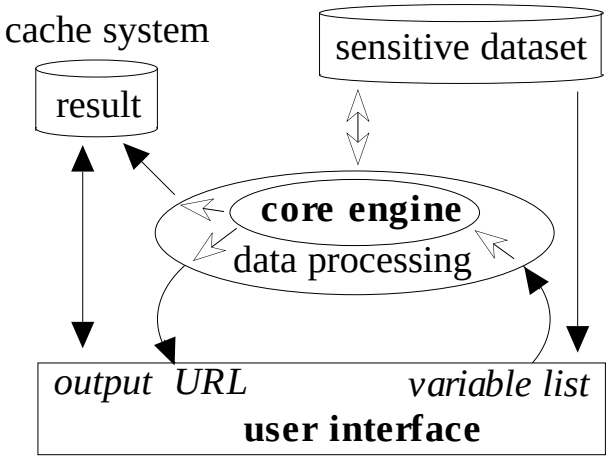

Figure 1: Outline of the data analyzing software with CGI

these conditions following above. Originally, these softwares are used individually. In this case, we use it as an external command. In other words, this is a "**recycle**" of the Fortran program on the CGI program via Web. It is expected that the developer speeds up creating CGI program by combing existing function. We do not need to write all script in one CGI program if we recycle the existing function.

There is another study by using CGI program to statistical analysis; Yamamoto and Tarumi[2] presented the system that can summarized and made tabulated data or statistical table for any variables in a home range survey at Okayama Prefecture. There is similar study by mean of re-using the Fortran program; Takeuchi[3] provided statistical DLLs based in Fortran programs to Microsoft Windows plat home.

2 Data analyzing system with CGI

In generally, Data analyzing software is consisted of mainly two parts: (1) User interface. (2) Core engine program for analysis. There are two purpose of user interface, one is to command how to analyze data to core engine, the other is to show the analyzing result on the interface. The purpose of the core engine is to receive the data and to analyze them. In past day, we could not analyze data in one system without interface and statistical engine. But, the Internet has increased and the concept of network has spread in the world. This situation causes that the two parts are not always necessary in one system. In addition to this, the variety of data transfer method from client to server, and that of the language used by development system cause multiplicity of analyzing softwares(Table 1).

The outline of our system are shown in Figure 1 and our system type describes as follows; (A) Data transfer method is via database. (B) Fortran (core engine), Perl (CGI program), Java (user interface for scatter plotting) are used. (C) Core engine is on the server. (D) User interface is a Web browser. (E) Access method between the interface and the engine is network transfer. In our system, a user can only show

Table 1: Various types of the Data analyzing softwares : (A) Data transfer method to engine, (B) What Language are used, (C) Which CPU is used at start core engine, (D) User interface type, (E) Accecess method between interface and engine.

(A)	(B)	(C)	(D)	(E)
upload	C	on server	Web+CGI	stand alone
down load	Delphi	on local	Web+Applet	network
via database	Fortran	decentralized	MS-Windows	hybrid
	Java		terminal	
	VB		X-Windows	
	R or S		hybrid	

a list of the names and descriptions about each variables in the dataset. The user can specify some interesting variables on the interface for reading these information, the submitting with these specified parameters to the server start calculation.

There are three reasons why we develop the system: (1) We process the data with Perl script at a data cleaning and extracting. (2) But, student can not do that on Unix system easily. (3) And also we treat sensitive raw data, so we can not give it to everyone. Often, the disclosure of raw data has legal limitation, for example, there is a contract between person and company on survey. The contract allows us to open only the result of survey, but dose not allow to open the raw data directly.

It is possible that it is analyzed on the web without open these data on the public. Web system has space that can not access from the Internet. In those circumstances, We put the data file on this hidden space or not on the server in order not to access the file directly. But, it is not necessary to worry that the external program do not access the data file, since this program and httpd daemon run individually.

However, we have to be careful about a back door trap of a system such that a symbolic link to the datafile put it in web accessible space. And also, The users on the same server can read the data file easily. To prevent from these situations, we should only register the user who has a permission to read the data file on the server. Or, we should set the file permission to be read only by the external program and also CGI program too. There are possibilities to be invaded by hacker or cracker. To prevent from this invasion, It is natural to set on a server in strong fire wall. In case we have the data stolen, we should encode the data. but it should be also considered that is takes a lot of time to decode on extracting.

3 The details of the system

First, we must explain about the data which were used here in little. The data which is used in example is presented from NTT data system lab. surveyed about finance into personal life in 1994. The data contain two type of question. (Type I) "How do you think about H?". (Type II) "Are you using F_i?, What is the reasons why F_i is being used? Choose as following : $G_1, \ldots, G_k$.". Type I question is completed

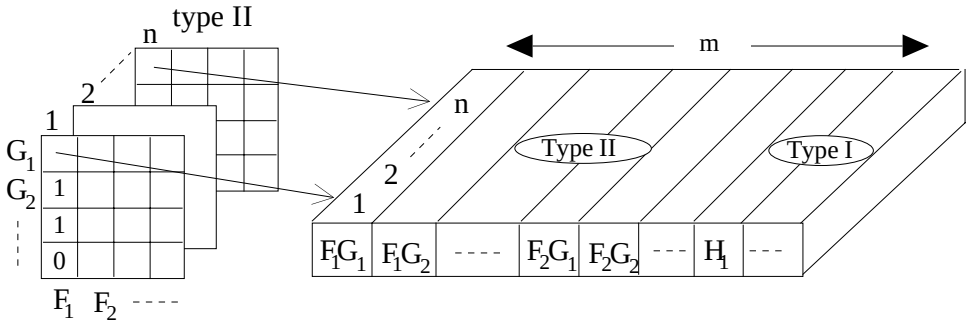

Figure 2: $n \times m$ square dataset : each variable has a unique name in database and contains n samples. Variable name, for example, F_1G_1 is coded A130101

by itself. and Type II question is related closely. For example, The F_i is "Sanwa Bank, Ltd.". G_1 is "to saving a marriage fund", G_2 is "to saving expenses bringing up children", and so on.

The value of Type II is defined by 0 (No) or 1 (Yes). These data are stored into the database mixed II with I (Figure2).

The meaning that apply INDSCAL model, it is one of the MDS procedures for INDividual differences SCALing[4], to these type of data is not concerned in this example. But, an explanation about INDSCAL is little necessary to know what CGI program dose and what is need to run INDSCAL program. The program requires l distance matrices. It is available that lowerhalf of similarity matrix without diagonals, full symmetric dissimilarities, and so on. We fix that the CGI program produces Euclidean distances lowerhalf matrix without diagonals. The ij-th element of a l-th distance matrix $(d_{ij})_l$ is given by following formula.

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik \cdot l} - x_{jk \cdot l})^2}$$

where $x_{ik \cdot l}$ is the occurrence of "yes" in the data F_iG_k which is categorized by a condition $H = l$. This is a distance between object F_i and F_j on p dimensional Euclidean space spanned by $G_1, \ldots, G_p$.

Our system is available on the web, the interface is shown in figure 3. There are three parameters we should input. First parameters decide objects F_i that are plotted as a point in stimulus space. Second parameters construct vectors of the farmer objects in p-dimensional space. p is number of checked box in second parameters. Third parameter need to divide samples into some category for obtaining weight space for viewing the difference between in it. The user gives a name of variables in the left box directly. It is available to change category for input the string which indicate rules in right box. For example, Suppose that there is 10 categories in variable "Q12", Type `Q12` in left box and the string "`1:5->1;6:10->2`" in right box, Then the categories is divided into 2 categories $\{1,2,3,4,5\}, \{6,7,8,9,10\}$. These is also a link to see variables list, which is summarized in excel file on the server. When

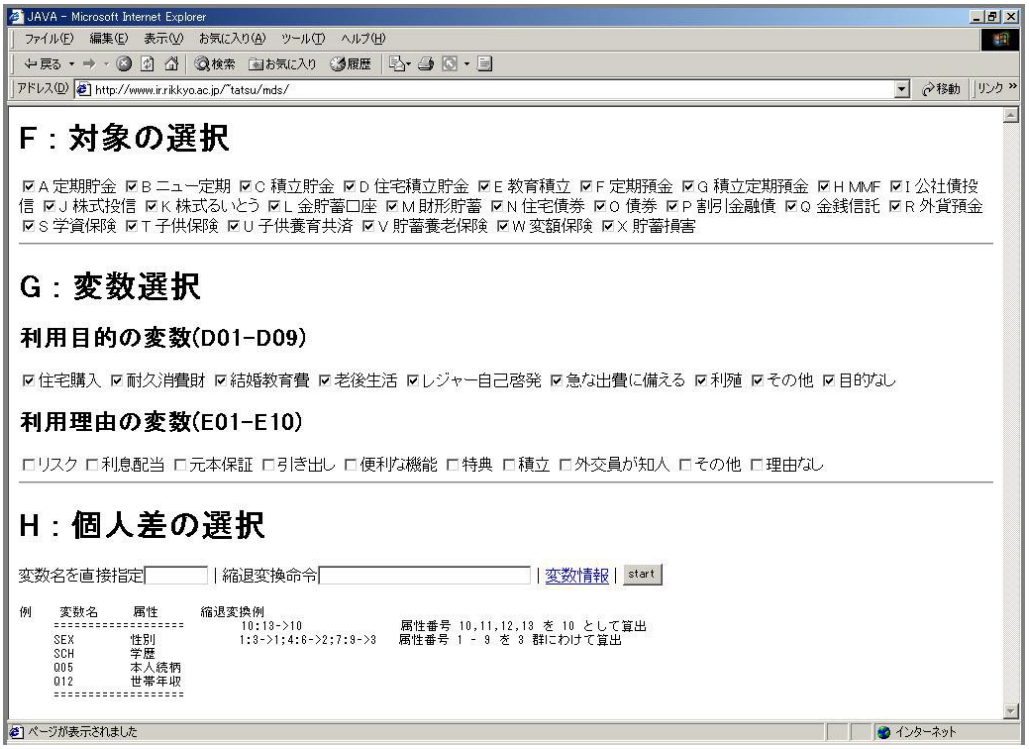

Figure 3: A screen shot of the input interface

all parameters were decided, the user submit these information to the server for pushing a "start" button. The interface is written in HTML with FORM tag. Each selectable parameter has unique name. In addition that, There is some small trick in it. that is, the hidden parameter is involve in the HTML. `<INPUT TYPE="hidden" NAME="object1" VALUE="A_01:18;B_01:06">`, This hidden parameter gives a range of parameter name, and it is used at test in CGI program whether parameter `A01` which is `<INPUT TYPE="checkbox" NAME="A01" VALUE="A">` is selected or not.

On server side, CGI program is running as follows; (1) check parameters: if no parameters found in variable list then return back to error message to user. (2) make a similarity matrix from the farmer parameters and the database: There are also some trick to generate ID name in database, `<INPUT TYPE="hidden" NAME="rule2" VALUE="AD=A13;AE=A14;BD=A16;BE=A17">` denote that if selected both `A01` which is in the first parameter F and `D01` which is in the second parameter G, then the string `A01D01` expand to `A130101` by CGI program. e.g. both `B03` and `D02` expand to `A160302`, and so on. (3) call the Fortran program: `system(./sindscal < input > output)`, `sindscal` is the Fortran program it was already compiled by G77. Both input file and output are controlled by process number. (4) finding and extracting the result from the previous output file: Pattern matching system in Perl language help us to do it. (5) backing results to the user interface: CGI program return HTML file which involve the locations of (a) all logging file while running, (b) Java program for plotting points of stimulus space in two dimensions with also the location of the

plotting information. (6) CGI program was terminated.

Again on client side, Web browser load Java Applet from the web and run the plotting program on the client machine. The program load also the plotting information from the Web by setting up a parma tag as follows; `<param name="datasrc" value="http://www.ir.rikkyo.ac.jp/~tatsu/mds/dbase/pdata.<!--dataid-->.dat">`. Here `<!--dataid-->` is replaced to own process id number by CGI program at the previous step. This Java program draws only points and label in now.

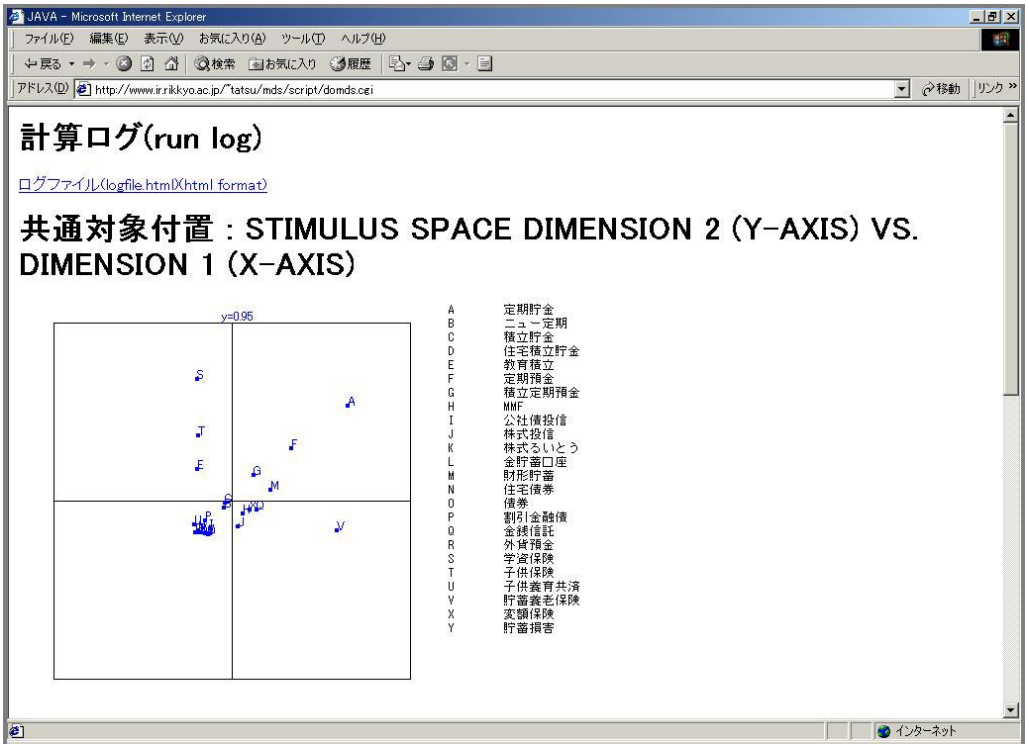

Figure 4: A screen shot after calculation

References

[1] Netlib_maintainers. Netlib Frequently Asked Questions. Available via Internet from `http://www.netlib.org/`

[2] Yamamoto, Y. and Tarumi, T. (1997). An application of microdata on the Web [in Japanese]. *Proc. 65th Annual meeting of Jpn. Statist. Soc.*, 160–161.

[3] Takeuchi, A. et. al. (2000). Dynamic link library for statistical analysis. *Proc. Comp. Satatist.*, Short Communications and Posters, 267–268.

[4] J. D. Carroll and S. Pruzansky (1970). Three-way scaling and clustering. *Psychometrika*, Vol 35, 283–319.