A TEACHING EXPERIENCE THROUGH THE DEVELOPMENT OF HYPERTEXTS AND OBJECT ORIENTED SOFTWARE.
Proceedings del Convegno "NGUS'95 III International meeting: new trends in theory, software and application of multidimensional data analysis" editi a cura di F. Mola (Univ. Di Napoli) e stampati dal CISIA di Parigi.(1995)

A teaching experience through the development of hypertexts and object oriented software.

Marcello Chiodi
Istituto di Statistica. Facoltà di Economia-Palermo-Italy

1. Introduction[1]

This paper is concerned with some recent teaching experiences, still in progress, for which I developed software in graphical environment for few specific topics (besides the main tools of standard statistical software) as an aid to my theoretical lessons.

Furthermore, in the last year I developed a small *hypertext* about my course of *Statistica Matematica*, running in the Windows environment, which has been improved in the present semester of teaching.

The key idea was: I develop software for my researches and I use statistical software; I teach statistics; then, I tried to use my software and programming knowledge to improve my teaching.

The impact on teaching has been quite impressive, because use of software in graphical environment can give a good visual aid to the understanding of even complex theoretical topics, and can give also a dynamic dimension which cannot be given with standard tools: running a true simulation with a video projector, with the display of many simulated regression lines, at least is better that trying to describe it on a blackboard, or watching at a table of numbers. Also it allows to introduce theoretically complex subjects by visualizing the relevant concepts.

---

[1] This work has been supported by M.U.R.S.T. funds (60%) in the framework of a research project of Prof. M.Chiodi

The present paper is intended as a report on this experience which seems to have a valuable impact on teaching; however the software and the hypertext is conceived as an aid, and not as a substitute, to standard tools, e.g. text books, lessons, lectures, exercises, standard statistical software, etc.

This paper is so organized: in the second section the developing of an hypertext is presented with the different possibilities of use during a regular course of lesson. In the third section some application of object oriented software to specific topics is exposed. In the fourth section the use for teaching of a more developed software on statistical simulation is sketched; finally in the fifth section some details on the languages employed for this software is given, together with some future directions of this work.

2.Development of an hypertext

The idea of the development of an hypertext arose during my course of *Statistica matematica* at University of Palermo, for students of the degree in Statistical and Economical Sciences. At the beginning it started as an experiment of collecting and organizing notes and formulae drawn directly from my slides used in the lessons. Later, also with the encouragement of my students, I improved the hypertext, so that it could be part of the didactic material of this academic year; it is still an *experiment*: anyway it is an hypertext running with a standard graphical interface (supported also with graphical objects obtained with other standard applications) conceived simply as an aid to my lessons and I am using it also as didactic material for my students. However *it is not thought as a stand-alone product* and *it is not meant as a substitute for a text book*, neither as a computer aided self-instruction tool.

Nevertheless, the present teaching experience, besides the insertion of new paragraphes and some suggestions of my students, is showing that sometimes the students actually use this hypertext in a different and often more advanced way, with respect to the way strictly intended by me, because the hypertextual structure enables a lot of routes not necessarily known in advance at the moment of the software development.

2.1 Hypertext structure

At the beginning of this experience, there was a lot of work in order to create the structure of the hypertext: the main structure is hierarchical, with chapters and paragraphs, but not very well structured as a full hierarchy, because it was built to be used with my lectures and not as a substitute for a book: the main aim was to collect some formulas and formal rules. Teaching

Fig1. Differences between hierarchical and hypertextual structures

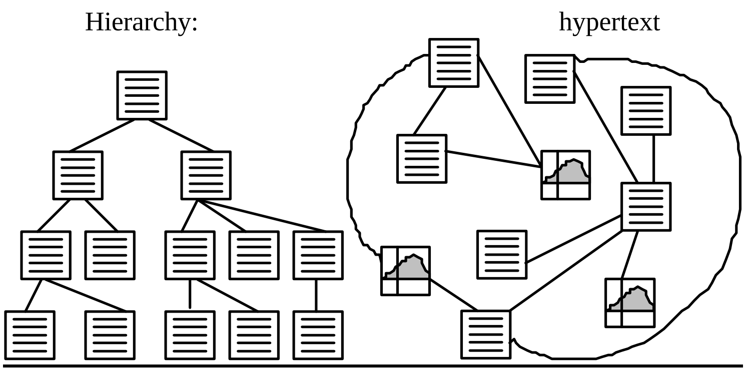
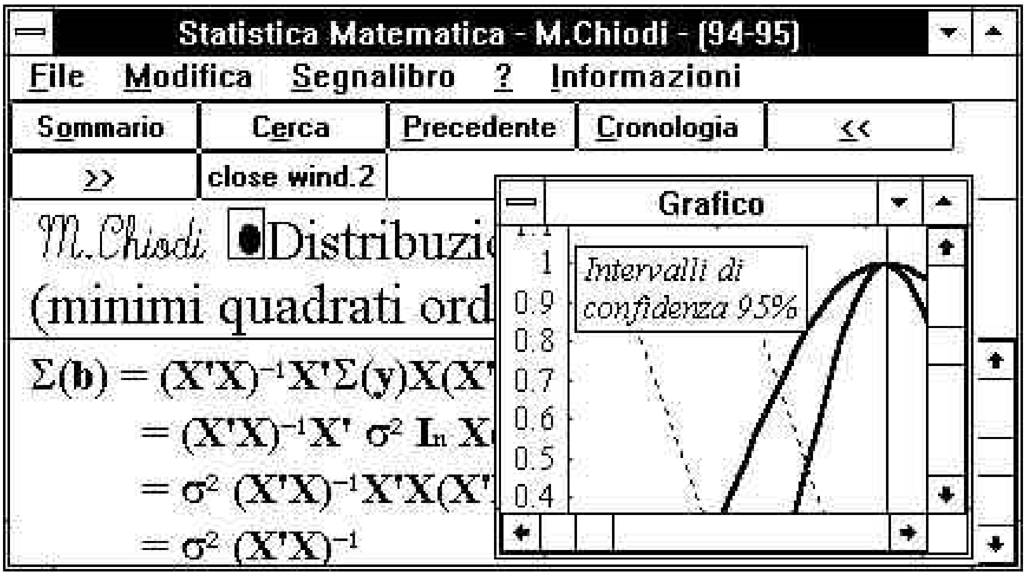


Fig.2 Sample display of the hypertext



material is often made up by miscellaneous and non sequential papers, or graphics, exercises and so on: so, using an hypertextual structure, it was easy to link together the different topics and notes and to link them to the main hierarchical tree.

In Fig.1 graphical representations of hierarchical and hypertextual structures are sketched: the former is a typical tree structure, while the latter is similar to a network (Angelides, Tong, 1994; Nielsen, 1990). Hypertext can be tought as the representation of links between pieces of data of even different kinds: text, graphics and also audio, video, or other hypertexts.

Once the main structure had been set up, I found that the effort needed to add new topics, or new graphics or exercises, was very small, compared with the usefulness and the compactness of the results. Also the updating and the correction of text is straightforward. So there is the possibility of frequent updating of the teaching material, even simultaneously to the lessons; textual and graphical contributions (or exercises given to students) carried out during the practice courses and during the lessons are also inserted in the hypertext, so that it increases considerably during the semester of lectures. In Fig. 2 a sample display of this hypertext is shown. The main subjects covered are numerical methods in statistical estimation, departure from normality in general linear models and inference in non linear models.

2.2 Features of the hypertext (user point of view).

The main feature of an hypertext is the variety of different ways to access the different topics:

-sequential reading according to predetermined sequences of topics;

-hierarchical reading, according to a main tree structure, e.g. traditional book structure with chapters, sections, paragraphs and so on;

-*hypertextual* reading, that is jumping with marked hyperlinks from a topic to other logical connected topics; jumps can also point towards a glossary of short definitions, which can be very useful in order to maintain the reading of a main topic, with some notes;

-key words search, thanks to sequences of key words linked to each topic, which can be displayed alphabetically ordered;

-menu driven choice of options and subjects;

-interaction with other applications, simultaneous reading of more than one paragraph and so on with other standard hypertextual options.

Other facilities which forced us to choose this particular software are:

-printing pages, or copying pages to foreign applications;

-simultaneous displaying of more pages;

-reverting the order of reading, coming back to the most recently read pages;

-saving notes and bookmarks.

2.3 Use of the hypertext with students.

One of the main difficulties in the construction of the hypertext, is the meaningful insertion of hyperlinks, that is, the possibilities of jumping from one topic to another topic (e.g. in the topic on general linear models, the possibility of jumping to some brief reminding on basic matrix algebra). But the problem is, from the developer point of view, which words or phrases should jump to some other connected topic? Or, conversely, which are the point of the text in which the student would like, or need, to jump to some other topic?

I solved this problem trying to simulate lessons or real studying on some topics, or actually seeing the reactions of students while watching them using the software. Actually this has been one of the most difficult task, because teachers often can not known what will be the student's need of more information.

3. Object oriented software

I implemented software to show graphically some of the numerical methods employed in the solution of classical statistical problems: maximum likelihood estimates (MLE) of the parameters of some selected distributions.

It underlines the behaviour differences between Newton method and direct search method, through 2d and 3d display of the updating steps needed to find maximum likelihood estimates, together with the effects of the different choices of the initial point. The aim was not to have a professional software to solve minimization problems: for this purpose of course students are guided to the use of some standard programming tools, like non linear regression or weighted regression modules in advanced statistical packages, or like optimization subroutines libraries (e.g. IMSL libraries). The main purpose of developing some graphical software is only to give some detailed examples on few parametrical models with 2 or 3 parameters, about the problem of actually computing the MLE estimates, showing the numerical steps towards the solution as well as the graphical representation of the steps, by means of a sequence of lines (triangles for direct search simplex method) on a contour levels plot or on a surface plot: the visual impact is valuable, and I experienced that this is a good way to support theoretical lessons on numerical minimization, which can take a lot of time in a compact course, to give at glance the underlying ideas of the main methods for optimization. This teaching approach has been useful also to convince students of the truth of some statements found on books of statistical inference beginning with something like: "MLE estimates can be computed by equating to zero the derivatives,..." and ending with something like: "but *sometimes* the system of equations is difficult to be solved explicitely in explict form".

4. Software on simulation techniques

The first version of this software has been carried out for a workshop on simulation studies for the students of the doctorate on *Computational Statistics* and then I used it for my course of *Statistica Matematica.* The original aim was to show to students experienced in theoretical statistics the usefulness of simulation techniques in statistics, because they can approximate empirically (and then, visually) a lot of results that are already theoretically known, like sampling distribution of regression estimators, or comparison between *true* and *observed* straight regression lines.

Successively, I discovered that I could invert the teaching process in basic statistic courses, that is, I introduced a certain number of important theoretical results by showing them by simulation, for students who were not trained on theoretical statistical inference: in fact simulations are often intellectually simpler than formulaic methods (Simon, 1994), because they directly *show* sampling distributions, without calculating them theoretically (by integration, or by characteristic functions and so on).

4.1 Some details on the simulation used for teaching

The software is divided in two main sections: elementary simulations and simulations of linear and non linear regression relationships.

I employed tested multiplicative congruential generators (Downham, Roberts, 1967; Golder, Settle, 1976) to obtain uniform pseudo-random numbers

U_i, which are used by all the other routines to obtain random numbers from some discrete and continuous distributions; the algorithm uses the recursive rule:

$$L_i=\lambda L_{i-1} \bmod c;\ U_i=L_i/c;$$
$$\text{with } \lambda = 8192 \text{ and } c=67099547,$$

This choice of λ and c ensures a cycle of generators of lenght c, with the possibility every time of entering a new seed L_i. However the goodness of sequences of uniform numbers can be directly tested using the software, visualizing histograms of the simulated frequency distribution of uniform numbers, together with the Pearson's chi-square test on large sequences of numbers. This has been revealed a good tool to convince students of the goodness of the algorithms without giving the details of the software employed.

The first section shows visually some elementary simulation tools:

-drawing pseudo random numbers by inversion of the cumulative distribution function, with dynamic graphic of the transformation procedure;

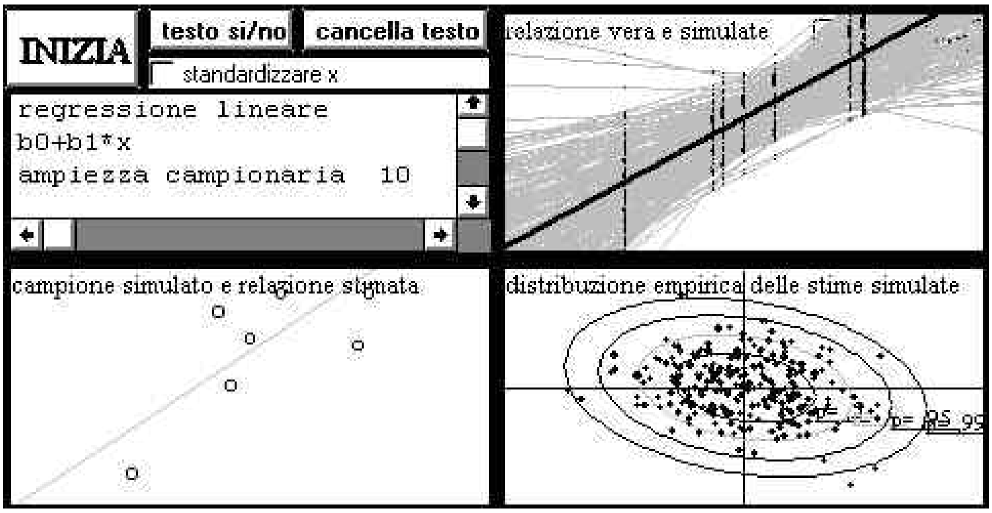
-numerical integration by Monte Carlo method, obtained by displaying n points with different colours according to the fact the they fall below or over a function to be integrated: a glance at the display after the simulation, gives immediately the idea of the approximated integral as the empirical relative frequency of points below the integrand function. The display is accomplished by a comparison between true and estimated value of the integral for increasing values of n;

-simulating the drawing of samples of uniform, univariate and bivariate normal, and normal of order p variates (Chiodi,1986): display of simulated and theoretical distributions and randomness test, with the possibilities of displaying the 2D and 3D histograms step by step;

-simulations on the central limit theorem: sampling distributions of arithmetic means of samples of any size from different parent populations are simulated, and compared with the proper normal limiting distribution: the qualitative behaviour of the central limit theorem can so be appreciated, together with the goodness of fit of the normal limiting distribution, depending on the different sample sizes and different parent distributions, symmetrical or not.

I found the last option very useful for statistic courses for non statistician students (e.g. applied sciences students as medicine students, biologist), because central limit theorem is one of the basis of inference, but is very hard to be shown analitically because this involves advanced mathematical and probabilistic tools: it's hard to request this knowledge for students of a short course.

fig. 3 Sample display of the simulation software



4.2 Simulation on linear and non linear regression.

In the second section of the software there are the more used and developed simulations: there is a starting option panel, in which many options are selected for the generic (univariate) regression model:

$$y_i=f(x_i,b)+\varepsilon_i \qquad (i=1,2,...n)$$

The choices at disposal are:

- the number m of samples simulated
- the number n of units for each sample
- the functional form: linear or some non-linear model (exponential with different parametrizations, logistic);
- the *true* values of β_1 and β_2 (I chose, when possible, models with two parameters only);
- the n design points x_i: arithmetic progression, or randomly chosen, or given at run-time or read by external text files;
- choice of the parent distribution for the random error ε_i: if gaussian or normal distribution of order p and choice of the scale parameter;
- choice of the estimation method: ordinary least squares, or L_p norm.

After some experiments of different possibilities of displaying the results of the regression simulations, I obtained the clearest results by dividing the screen in four quadrants, as shown in fig.3: in the first one only numerical results are displayed, in the second quadrant the *true* regression relationship (linear or non linear) is displayed together with the fitted *simulated* relationships; in the third quadrant a simulated sample, together with a fitted regression, is drawn; in the fourth one, the empirical sampling distribution of the estimates of two parameters is plotted, together with the ellipsis of the proper asymptotic normal distribution.

The main feature of this section is that simulations can be carried out step by step, generating each sample with a mouse click, or automaticacally generating all the m samples, so that different lines and curves begin to run on the 2nd quadrant plot, on the 3nd quadrant plot samples are drawn until the number m of total samples is reached and finally, on the fourth quadrant the m estimates of the parameters are plotted.

In my experience of teaching statistical inference, I found that the concept of sampling distribution in regression is very hardly understood by beginner students, because it is not easy to justify some key concepts, like "x regressor are fixed, but y is randomly distributed with expected value f(x,**b**)...". Furthermore the different distribution of the estimated values related to the distance from the average of the x, can be easily understood, by watching the set of simulated regression lines. The simultaneous parameters estimates distribution is visualised and overimposed on the plot of the normal ellipsoidal sections: so the departure from normality for small values of n of the sampling distributions of the MLE for non linear models can be easily appreciated.

5. Software details and future directions.

At the present time, the software used to develop the hypertext is the Microsoft help compiler for Windows, which produces standard Windows help file with hypertext structure under the environment of Windows 3.1. For the object oriented software I used mainly the Microsoft Visual Basic 3.0 for Windows for the graphical interface, linking some well tested numerical routine, written either in Visual Basic or in FORTRAN. In the next months I will write the whole application for the Windows95 environment: this will enhance the graphic facilities, enabling more complex 2D and 3D plottings.

The more important thing however is that the teaching experience enables me to develop and improve only sections useful for teaching, which really helped student in understanding complex subjects, avoiding to develop sections which have a poor impact on teaching.

References

ANGELIDES M. C., TONG K.Y.(1994), Using Hypertext in Developing the Human Computer Interface to Gaming-Simulation Environments that Incorporate Intelligent Tutoring Support, *Journal of Computing and Information Technology*, 3, pagg. 21-34.

CHIODI M. (1986), Procedures for generating pseudo-random numbers from a normal distribution of order p (p>1), *Statistica Applicata*, 1,pagg. 7-26.

DOWNHAM D.Y., ROBERTS F.D.K. (1967), Multiplicative congruential pseudo-random number generators, *Computer Journal*, 10, pagg. 74-77.

GOLDER E.R., SETTLE J.G. (1976), The Box-Muller method for generating pseudo-random normal deviates, *Applied Statistics*,25,pagg. 12-20.

NIELSEN J (1990), *Hypertext and Hypermedia*, Academic Press, USA.

SIMON J.L. (1994), What Some Puzzling Problems Teach About The Theory of Simulation and the Use of Resampling, *The American Statistician*, 48, pagg.290-293.

Summary

A teaching experience through the development of hypertexts and object oriented software.

In this paper some new teaching experiences on statistic topics are presented. They concern the development, during my regular courses, of a small hypertext and object oriented software, mainly on statistical simulation. The hypertext resulted to be quite useful during the course, in order to support the ordinary teaching material, like slides, graphics and so on.

The object oriented software gave me the chance of introducing some hard or theoretically complex topics only by a graphical approach: the simulation software has been used to visualise simulated sampling distribution,

in univariate samples and in linear or non linear regression; this resulted useful in lessons for advanced courses on statistics as well as for introductory courses on statistical inference, where there is the need of giving some basic ideas on statistical inference, without writing many formulae.

Keywords
Teaching, Hypertexts, Software, Simulations, Numerical Methods.