

E-STATUS: A WEB TOOL FOR LEARNING BY DOING EXERCISES

José Antonio GONZÁLEZ

Statistics & Operations Research Dept.

Universitat Politècnica de Catalunya

Fac. de Matemàtiques i Estadística, C/. Pau Gargallo 5, 08028 Barcelona Spain

Jose.A.Gonzalez@upc.es

Pilar MUÑOZ

Statistics & Operations Research Dept.

Universitat Politècnica de Catalunya

Fac. de Matemàtiques i Estadística, C/. Pau Gargallo 5, 08028 Barcelona Spain

Pilar.Munyoz@upc.es

ABSTRACT

The paper introduces the project led by a team of teachers to assist students learn statistics. The goal is to build a tool able to present mathematical problems and to correct the students' answers. The problems may include random data, so the solution cannot be previously known (if solved before) and the student can reconsider it if necessary. Pedagogical implications are commented, since the method can be effective on the basic and middle domains of learning, as well as on higher levels, specially if careful design of the problems is applied.

1. Introduction

The objective of this work is to present a web-based tool which allows the student to solve probability and statistical inference exercises in the introductory course taught in 2nd year degree in Computer Science Engineering at the Barcelona School of Informatics (FIB) of the Universitat Politècnica de Catalunya (UPC).

Our teachers (eight to ten involved) are teaching a large introductory statistical course to over 250 students. The course usually lasts 15 weeks (one semester) and the student attends six weekly hours of statistics sessions. Students are divided into groups of 80 students or smaller groups for the work in the laboratory. (The student's profile belongs to three different Computer Science degrees by the same university).

We agree with the opinion expressed by Roberts & Simonyi (1997): "A major challenge in teaching introductory courses to a large, diverse audience is the wide variation in background and ability that exists in the undergraduate population, which makes it hard to find the appropriate level of instruction." It is for this reason that we have thought of a product that allows the weaker students to practice many times the more difficult concepts without obstructing the progress of the advanced students.

It is well known that introductory statistics courses require important individual work on the part of each student, who needs exercises and practical cases in order to use the concepts acquired in a very short period. Exercises are undoubtedly a good complement for the "theoretical" lessons: they show how the concepts explained in the classroom are applied in real or simplified cases, and they facilitate the comprehension of the exposed ideas, through a typical process of learning by doing.

Why have we designed a web-based tool? The reason is obvious. Although our students are non-distance learning students, they will have a tool available to practice the statistical problem resolution, very accessible, and it could be used any time, at any place. The only condition is having a computer with Internet access. The difference between our project and most web-based tools (the list of examples would fill many pages) is that our software is dynamic in the sense that every execution presents new sample data randomly generated. The student could do the same exercise again but the data (and the solution) would be different.

The proposed methodology can be useful to many other courses on a mathematical basis, mainly in engineering studies, whose students make extensive use of problem solving in order to lessen the level of abstraction present in the classroom.

2. Motivation

We agree with Garfield (1994): "As goals for statistics education change to broader and more ambitious objectives, such as developing statistical thinkers who can apply their knowledge to solving real problems, a mismatch is revealed between traditional assessment and the desired student outcomes. It is no longer appropriate to assess student knowledge by having students compute answers and apply formulas, because these methods do not reveal the current goals of solving real problems and using statistical reasoning". The development of mathematical thinking in our students is capital for their professional future: their ability to manage a problem and lead it to an efficient solution is highly related to their analytical abilities. However, students will achieve an appropriate level of mathematical reasoning only if they face up to different situations compelling them to apply their higher aptitudes and, hence, reinforcing them. These aptitudes, jointly known under the name of *intellectual habits*, include several levels: 1) comprehension, 2)

application, 3) analysis, 4) synthesis and 5) evaluation, and are related to the ability to transfer the knowledge from one field to another, the competence to face up new problems, and in general to what is called critical or reflexive thinking. These levels, based on Bloom's taxonomy, can be revised in Besterfield-Sacre et al. (2000).

The assessment methods used in our statistics course are, Muñoz, P. et al. (2000): 1) one test/quiz after the first 7 weeks including calculations and basic questions, 2) exercises delivered every two weeks and marked in the classroom, 3) one project developed in groups with data simulated from a realistic situation and covering a broad range of course objectives, and 4) the final exam. Obviously, this tool is a complement to all the learning materials used in it.

3. Present framework and further work

Using a computer application able to generate individualised problems to students, getting their response and providing the correct solution is an old ambition for us, teachers of statistics courses in several university centres. A former prototype, *Autoproblem*, was developed by two students as a part of their career final project: it was designed to pose fixed questions about inference with one and two samples, mainly confidence intervals and test hypotheses. The data is randomly generated according to the specifications given by the teacher: the sample size, the probability distribution and parameters representing the population. The student runs a Java applet on a standard internet browser, connected to the remote server installed in our department, where the results are stored in files.

Although it was an interesting experience, it showed the drawbacks of a closed application: more code development would be necessary to include new thematic modules, e.g. ANOVA or linear regression. On the other hand, the management of the information obtained should be improved to a great extent.

e-status is the next proposal, designed to overcome the aforesaid deficits. In the following section we extensively describe the working mode of the tool, designed as its predecessor according to its different applications for the teacher and for the students. At present, the tool is being implemented and we expect it will be available by September 2002. However, we are planning a gradual introduction: first year students will use the tool experimentally, but we will not exploit it as an assessment procedure until we have reached full comprehension of its educational effectiveness through the experience.

Future expansion of the tool will greatly depend on the needs revealed by its use. There are some implications that have not been developed so far, but will as soon as enough resources are available. Some of them are: definition of macros, in order to avoid repetition of code and auxiliary symbols; inclusion of external functions, which will be computed by other application; enriched grammar, now limited to constant strings, logical and numerical expressions, which can be extended to consider graphic objects, very interesting from an educational point of view.

4. Description of the environment

What is a problem?

Usually, in the educational world we think of a problem as a (real or imaginary) situation and a number of unknowns that can be deduced from the explanation. The goal is to find out how students apply their knowledge and reasoning to find the solution.

Many mathematical problems possess a solution that can be deduced analytically: this is the kind of problem considered in this work. From now on, a *problem* is an object consisting of:

- A Situation: a text describing the case
- Formulae: a set of equations according to the structure $symbol = expression$
- Data: known parameters of the problem; they are in fact a subset of the symbols appearing in the formulae
- A Quiz: a set of questions addressed to the student, each one composed by a text and the symbol giving the answer
- Metadata: additional items related to the problem, e.g. author, date of creation, lesson, title, difficulty, etc.

Example:

Situation	A farmer wants to know the area of his rectangular field. He measures the length of two sides, X and Y.	
Formulae	X = 120	
	Y = 80	
	A = X*Y	
	P = 2*(X+Y)	
Data	X Y	
Quiz	Which is the area of the field?	(A)
	What length of fence will the farmer need to enclose the field?	(P)
Metadata	title: the farmer's field; lesson: the rectangle	

How do the students solve a problem?

Let us define an *exercise* as a solution provided by a student to an instance of a given problem. We call *instance* of a problem a particular presentation of the situation, the data and the questions present in the quiz. That is, taking the previous example, the instance would be something like this:

The farmer's field	
A farmer wants to know the area of his rectangular field. He measures the length of two sides, X and Y.	
	X = 120
	Y = 80
1. Which is the area of the field?	(___)
2. What length of fence will the farmer need to enclose the field?	(___)

A good exercise would obtain 9600 as the solution for the first question and 400 for the second. Logically, all the instances of this problem are identical, and this could not be interesting for teachers. But they can change the problem definition:

...	
Formulae	X = Uniform(1, 100, 140)
	Y = Uniform(1, 70, 90)
...	

Now, each instance gives random values for the parameters X and Y, and the correct answer can not be known in advance: only values matching the symbols A and P are good.

Architecture

The teachers' application

A computer application, written in Java, is used for the manipulation of the problems. It allows:

- Management of files: creating new problems, saving them or retrieving them from a database, printing, etc.
- Edition: each component of the problem is edited assisted by the program
- Testing: the author can verify the correctness of the case by creating an instance

The person editing a problem ought to be familiar with the syntax of the expressions appearing in the Formulae section. The grammar defined for the expressions is far from being strange; on the contrary, we have tried to make it as close to standard as possible. Some of its features are:

- Common types considered: integer and real numbers, boolean expressions (true, false), strings of characters; for numbers, we have constants, vectors and matrices.
- Common operations and their precedence order: exponentiation, multiplication, division, addition, subtraction, etc. For the boolean type: negation, logical and, logical or.
- Arithmetic functions: trigonometric, square root, log and exp, etc.
- Special functions: like vector or matrix processing; special attention to the family of probability and statistics functions (random generation, probability distributions, etc.)
- Some functions and operators can be overload, that is, they can manage either constants or vectors, or even matrices, if the operation is allowed and the result is well defined.

Let us see some examples of possible expressions:

Formulae	N = 10	
	X = Normal(N, 100, 10)	sample from N($\mu=100, \sigma=10$)
	Mean = Sum(X)/N	Sum returns the sum of x_i
	Var = Sum((X-Mean)^2)/(N-1)	constant subtracted to vector
	Stdev = Sqrt(Var)	
	Alpha = 0.05	
	T = Invcdf(T, N-1, 1-Alpha/2)	evaluates inverse of a CDF
	R = T*Stdev/Sqrt(N)	
	CI_l = Mean - R	
	CI_u = Mean + R	
	OK = CI_l < 100 and CI_u > 100	μ inside the interval?

The previous formulae show how we can obtain the 95% confidence interval for the poblational mean with a sample (that should probably be given to the students as data of the problem).

The students' application

Any student enrolled in a course could access to the application entering a web address supported by the department web server. They have to authenticate themselves with their identifier and password and, once verified that the student is enrolled in the specified course and the validity of the identification, they can:

- Pick a problem in order to practice freely and enhance their autonomous learning
- Choose a block of problems (they cover different lessons and are like an ordinary exam)

- Take an assignment, maybe as part of the assessment process
- Monitor their own results until that time

The program shows an instance, probably generated with random data, of a problem and waits for the student's answer. Then it informs the student of the result: correct and wrong responses, time spent and a score to provide an estimate of his/her achievements. Sometimes, a student makes an error, and carries it along to the next questions. Depending on the strictness of the correction (that can be specified as metadata in the problem), dependencies on the responses can be taken into account or not, as these dependencies can be deduced from the set of formulae. We think that allowing this "soft" judgement (that would lessen the penalty of carried errors) is valuable: students will be aware, on the one hand, that they are "fairly" assessed (and not "coldly", as one might expect from a machine) and, on the other hand, that their errors are not innocuous at all.

The Database and its administration

Students, problems and exercises are some of the entities involved in the process. In order to obtain the best possible performance with the operation of the data (mainly searches, input and output), they and their relationships are managed by a Database Management System (like SQL Server). Both the teachers' application and the students' application communicate with the Database remotely, although the teachers will work through a local network and the students' application is designed from the base to work using Internet.

Obviously, some functionalities are included to execute normal operation. They are only allowed to teachers or authorised persons:

- Creating and deleting a course
- Loading a set of students in a course
- Inquiring about a student
- Grouping several problems into a block
- Defining assignments; usually the teacher specifies a timing: when the problem(s) will be available to the students (e.g., the week from May 11 to May 18)
- Collecting statistics or assessments, useful for the evaluation

Moreover, the administrator has to consider profiles other than the students and teachers': for instance, teachers not involved in a course but interested in its materials, or general guests from anywhere accessing via web.

5. Success factors

The task of the teacher is, broadly speaking, to provide the students with a basis of knowledge, to stimulate their learning and to use suitable tools to measure their performance and achievement of learning goals. Taking into account that some goals can be related to the higher levels of the intellectual domain, requiring thus a direct interpretation of the teacher, the method we present can be clearly useful to the stimulation of the students and their involvement with the subject of the course. Moreover, by returning a score of the exercise promptly we reach an important objective in any educational field: to give the students (immediate) feedback of their progress, making of the evaluation an effective stage in the teaching/learning process.

To conclude, some simple pieces of advice are given. The sought effect is the implication of the instructors in the methodology so they do include it in their collection of educational resources, and preferably as one of the main ones. Their compromise is a key element to achieve a high degree of use on the part of the students. As the reader can see, some of the suggestions appearing

below are not different from general rules to compose good “traditional” problems, but we must insist on them in order to bear in mind that technology will not transform a deficient problem into an interesting one.

What teachers should consider

- Compose the wording and the questions accurately
- Write cases for each lesson in the course
- Present a variety of situations (leave aside your old bag of balls)
- Avoid to always use the same type of questions (“*compute the mean of the sample*”)
- Include different degrees of difficulty
- Remark the connections with the course lessons, and organise a suggested sequence
- Try to pose questions referring to each knowledge plane, not just those related to “mechanical” skills
- Elaborate good sets of problems (go over the course contents throughout every lesson)

What students should hear (and take into account)

- Alternate study and problem solving, following directions
- Spend the necessary time, don’t answer without thinking carefully
- Doing the same problem several times is advisable and useful, but you should also be aware of when you should change to another problem
- Check your progress regularly, and work to improve the weaker points.
- If you don’t understand the questions, or you are repeatedly wrong, look for a way out (e.g., see your teacher about it)

6. Conclusions

This work has to be logically continued with the analysis of the results obtained. Validating the method would be desirable, that is, we would like to say that the academic performance has clearly improved past methods. In fact, we know this kind of conclusions may not be drawn lightly, since one cannot isolate the many factors affecting the learning of the students. In any way, we have considered the organisation of reliable data stored in a database from the beginning of the project, and this point may be capital to achieve a satisfactory validation.

REFERENCES

- Besterfield-Sacre, M., Shuman, L.J., Wolfe, H., Atman, C.J., McGourty, J., Miller, R.L., Olds, B.M., Rogers, G. (2000): *Defining the Outcomes: a Framework for EC 2000*. IEEE Trans. on Engineering Education, v.43, n.2, May 2000
- Garfiel J. B. (1994): *Beyond Testing and Grading: Using Assessment to Improve Student Learning*. Journal of Statistics Education v.2, n.1
- Muñoz, P., González, J. A., Cobo, E (2000): *Information technologies in an advanced statistics course*. Proceedings of the Compstat 2000.
- Roberts, E., Simonyi C. (1997): *Encouraging Top Students in Large Undergraduate Classes*. Speaking of Teaching. Stanford University Newsletter on Teaching, Vol 8, No.2.