Stochastic Models Applied in Health Care and Medical Education

Augustin Prodan[1], Madalina Rusu[1],
Rodica Prodan[2], and Remus Campean[1]

[1] Iuliu Hatieganu University
Str. Emil Isac 13, 400023 Cluj-Napoca, Romania
(e-mail: {aprodan,mrusu,rcampean}@umfcluj.ro)
[2] MedFam Group
Str. Constanta 5, 400158 Cluj-Napoca, Romania
(e-mail: familiaprodan@yahoo.com)

Abstract. The paper presents a Java framework for stochastic modelling and simulation, used as an infrastructure to model and simulate real world activities, phenomena and processes, particularly in health care, patient monitoring and medical education. We modelled the flow of patients through medical units, considering both their arrivals and their stay in the hospital. Also, we implemented bootstrapping methods, which are quite useful in simulation studies. We created bootstrapping e-tools for simulating laboratory works and experiments, to be used in both didactic and research activities.
Keywords: stochastic model, distributional model, simulation, bootstrapping methods, Java framework.

1 Introduction

Previous research has shown that stochastic models are advantageous tools for representation of real world activities, phenomena and processes. Due to actual spread of fast and inexpensive computational power everywhere in the world, the best approach is to model a real phenomenon as faithfully as possible, and then rely on a simulation study to analyze it. Based on theoretical fundamentals in stochastic modelling and simulation [Ross, 1990], we implemented an object-oriented Java framework for stochastic modelling, analysis and simulation of problems arising in a practical context, particularly in medicine and pharmacy. We created an infrastructure, consisting of a collection of Java class libraries, which are used to model and to simulate distributional models, stochastic processes and Monte Carlo methods. The basic design philosophy of our object-oriented approach to simulation of the random variables by means of distributional models is presented in [Prodan *et al.*, 1999]. The object-oriented Java framework containing the set of baseline classes for stochastic modelling and simulation is presented in [Prodan and Prodan, 2001]. We implemented *bootstrapping methods* [Hesterberg *et al.*, 2003], then we created and implemented bootstrapping e-tools with the purpose of simulating laboratory works and experiments, in both didactic

and research activities [Prodan and Campean, 2004]. There are two reasons for creating such e-tools: (a) to reduce the number of animals (guinea pigs, frogs, etc.) used in experimentation (an ethical reason), and (b) to reduce the consumption of substances and reactants (an economical reason). Using a bootstrapping e-tool, the experimenter can repeat the original experiment on computer, obtaining pseudo-data as plausible as those obtained from the original experiment.

2 The infrastructure

We created an infrastructure consisting of a set of Java class libraries for stochastic modelling and simulation. The classical random variables are the simplest stochastic models, also called *distributional models*, which enter into the composition of other complex models. We propose a hierarchy of Java classes for modelling the classical distributions. Each distribution class encapsulates a particular $simValue()$ method (see Figure 1) incorporating a simulation algorithm, able to generate a specific value for that distribution. In other words, the simulation algorithms for distributional models are implemented via a polymorphic method called $simValue()$. The particular implementation in case of each simulation algorithm is based on one or more of the following techniques: the Inverse Transform Technique, the Acceptance-Rejection Technique and the Composition Technique (see [Ross, 1990] and [Prodan *et al.*, 1999]).

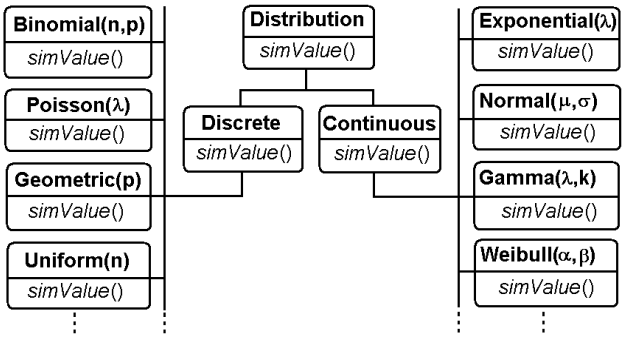

Fig. 1. The hierarchy of Java classes for distributional models

We consider three levels of simulation (Figure 2). The first level consists of simulating random numbers, as they are the basis of any stochastic simulation study. Using this first level, we build the second level, applied for classical distributions, for stochastic processes and for Monte Carlo methods. The third level of simulation is devoted to applications. As applications, we modelled activities, processes and phenomena from health care, patient

monitoring and pharmacy, we created e-learning tools (see [Prodan and Prodan, 2003] and [Prodan, 2004]) and we implemented bootstrapping methods [Prodan and Campean, 2004].

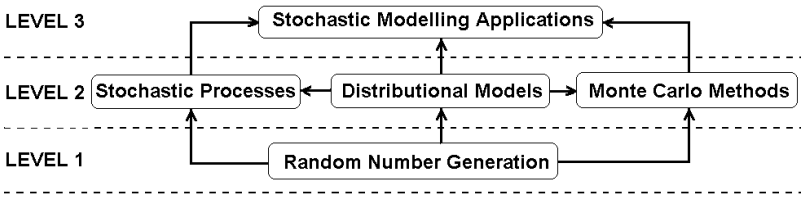

Fig. 2. Levels of simulation

To make a simulation study, it is necessary to generate more values, a sequence of values. One may choose to continually generate additional values, stopping when the efficiency of the simulation is good enough. Generally, one may use the variance of the estimators obtained during the simulation study to decide when to stop the generation of additional values. For example, if the objective is to estimate the mean value $\mu = E(X), i = 0, 1, 2, ...$, one may continue to generate new data values until one has generated n data values for which the estimate of the standard error (i.e. the standard deviation of the mean) is less than an acceptable value. We implemented a general simulation class as a *canvas*, which encapsulates the methods $doSimulation()$ and $doVisualisation()$. These methods are inherited by specific simulation classes for specific distributions (binomial, exponential, etc), which encapsulates specific $redraw()$ methods, able to show the results of specific simulations. Figure 3 shows the results of simulations from some discrete and continuous distributional models.

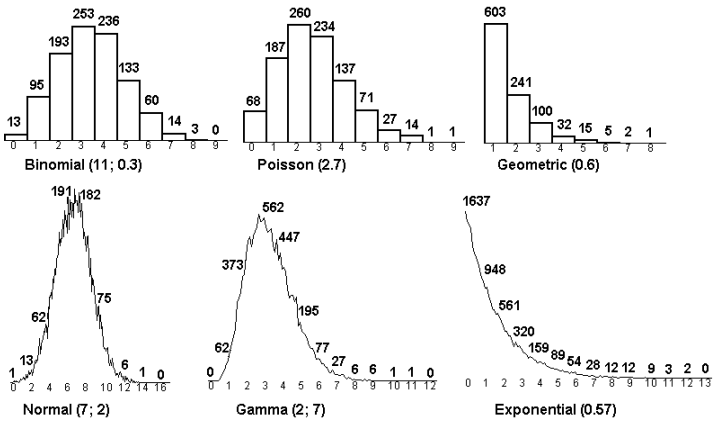

Fig. 3. Simulations from distributional models

When simulate from a continuous random variable X, a generated value $x \in X$ is approximated with a given *precision* expressed by the number of decimal digits to be considered. The user has the possibility to choose a precision of one, two, or more decimal digits. If no decimals are considered, the real value x is approximated by integer part of the x, the continuous random variable X is rudely approximated by a discrete one, and the results of a simulation can be graphically expressed in a segmented line format. If a precision of one decimal digit is selected, the results of the same simulation is more precisely visualized by a more refined segmented line. With a precision of two decimal digits, a more refined visualization is obtained. The higher this precision is, the higher is the *resolution* realized in visualization [Prodan and Prodan, 2002]. Figure 4 compares two visualizations for the same set of generated values from the exponential distribution with parameter $\lambda = 0.3$, the first visualization being with a precision of one decimal digit (Figure 4, graph a), and the second with a precision of two decimal digits (Figure 4, graph b).

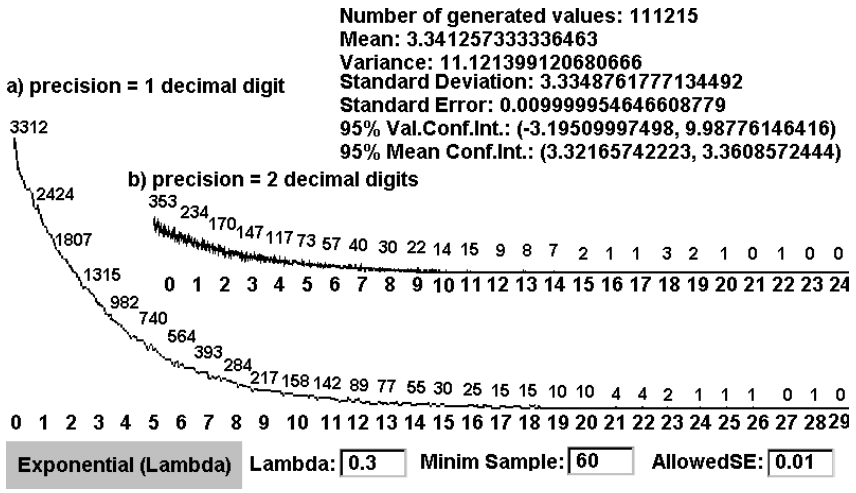

Fig. 4. Visualization with precision of one decimal digit, versus visualization with precision of two decimal digits, for the same set of generated values from the distribution *Exponential*(0.3)

As can be seen in this figure, when the precision grows with one decimal digit, the resolution grows ten times. With a precision of one decimal digit, ten numbers are considered between two successive integers, while if the precision is of two decimal digits, one hundred numbers are considered between two successive integers. When necessary, intermediate resolutions can be considered.

3 A model for patient flow simulation

Chronic patients may generally be thought of as progressing through two standard stages: firstly, the *acute care*, consisting of diagnosis, assessment and rehabilitation and secondly, the *long-stay care* where a small proportion of them remains in hospital for months or even years. Obviously, these patients may be very consuming of resources, situation which implies a serious analysis of health care costs in order to avoid the distortion of the performance statistics.

We applied stochastic processes to model the flow of patients through chronic diseases departments. The science of best designing the movement of patients through hospitals has not yet been discovered, but the use of the queuing theory models may provide a good enough solution to the problem. We intend to use results from both the queuing theory, particularly, and stochastic modelling, generally, in order to optimize the bed inventory and the cost-effectiveness of a hospital system. We describe patients arrivals by a Poisson process, hospital beds by the servers and the lengths of stay are modelled using *phase-type distributions*. In queuing terminology this is known as a $M/Ph/c/N$ queue, where M denotes Poisson (Markov) arrivals, the service distribution is phase-type, c is the number of servers (i.e. beds) and N represents the finite capacity of the system, comprising both waiting patients and patients being served [Gorunescu and Prodan, 2001]. It is also assumed that the queuing system is in steady state which, in practical terms, means that we assume that the hospital system has been running, in its present form, for a few years. This model enables us to study the whole system of geriatric medicine and is used to look either at the time patients spend in hospital, or at the subsequent time patients spend in the community.

In order to simulate the model, we have split it into two parts: the arrival of patients and the in-patient care. We modelled the arrival of patients as a Poisson process with a parameter λ estimated by using the inter-arrival times. These times are independent exponential random variables, each with the parameter λ and with the corresponding density function $f(t) = \lambda e^{-\lambda t}$. Figure 5 shows the results of a simulation, considering the arrival of patients as a Poisson process at rate $\lambda = 7.25$ patients per day.

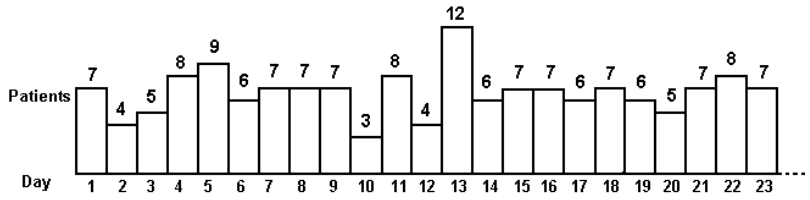

Fig. 5. Poisson arrivals at rate $\lambda = 7.25$ patients per day

The care time is modelled by the application of a mixed-exponential distribution, where the number of terms in the mixture corresponds to the number of stages of patient care. A common scenario is that there are two stages for in-patient care: *acute* and *long-stay*. In this case we compose two exponential distributions with parameters α and β, representing the access rates for the corresponding stages. The mixed-exponential phase-type distribution has the probability density function $f(t) = \rho\alpha e^{-\alpha t} + (1-\rho)\beta e^{-\beta t}$, which imply a mean care time of $\frac{\rho}{\alpha} + \frac{(1-\rho)}{\beta}$ days per patient. Figure 6 shows the results of a simulation with parameters $\rho = 0.07$, $\alpha = \frac{1}{77.18}$ and $\beta = \frac{1}{33.3}$.

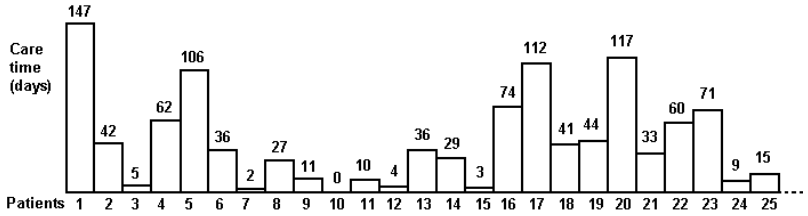

Fig. 6. The simulated results for in-patient care time

4 E-learning tools for medical education

Based on the infrastructure presented in section 2, we created e-learning tools and incorporated them into an e-learning environment, to be used by both the students and the teaching staff in their didactic and research activities. We implemented e-learning scenarios by looking at problems that can be put in a probabilistic framework. Every new concept is developed systematically through completely worked out examples from current medical and pharmaceutical problems. In addition, we introduced in each e-learning scenario specific probability models that fit out some real life problems, by assessing the probabilities of certain events from actual past databases.

As an example, we propose an e-learning scenario for students in Medicine. A learner that traverses such a scenario, will be able to apply a binomial distributional model in studying the chance of patients suffering from a particular type of cancer, to survive for at least a six month period after diagnosis. We would have to appeal to previous studies and information from actual databases to assess the chances of a patient surviving. This might indicate, for instance, that the probability of survival is $p = 0.3$, and consequently the complementary probability of death is $q = 1 - p = 0.7$. In real life, we are frequently interested what might happen to a group of patients we are studying. Therefore, we may formulate the following problem, as a piece of the current e-learning scenario:

Of the 11 patients in a particular cancer program, what is the chance of 7 or more of them surviving at least six months past diagnosis?

If p_k is the probability that k patients survives ($k \leq 11$), the solution is given by the sum $P = p_7 + p_8 + p_9 + p_{10} + p_{11}$. The distributional model $Binomial(11, 0.3)$ gives the values for p_k ($p_7 \approx 0.017$, $p_8 \approx 0.003$ and $p_9 \approx p_{10} \approx p_{11} \approx 0$), hence the solution is $P \approx 0.017 + 0.003 = 0.02$. The e-learning scenario may be resourceful in showing additional information about probabilities and statistics. We prepare and configure suggestive visualizations, based on a friendly and efficient dialogue with the learner. As an example, for any learner may be useful to see the previous probabilities p_k in a suggestive column format, and to recognize the solution to previous problem shown with dashed columns (see Figure 7).

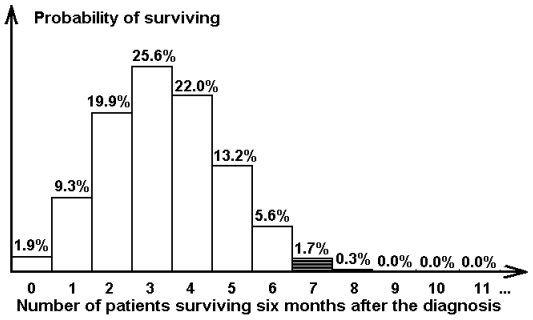

Fig. 7. A graphical solution for patient surviving problem (dashed columns)

An e-learning scenario combines simulation with interactive visualization and allows the learners to explore the knowledge bases with some well-defined learning purposes. We define a simulation class and a visualization class for each application object. These classes are then configured to obtain a particular simulation with a specific visualization. In an e-learning scenario, visualization is an active part of the system, serving as an additional interface for modifying dynamically some parameters. For example, the same distributional model $Binomial(11, 0.3)$ may be applied in an e-learning scenario for students in Pharmacy, studying the effect of digitalis on frogs. Suppose we know from previous studies and experiments, that injection of a certain dose of digitalis per unit of body weight into a large number of frogs, causes death of 30% of them. We may propose the following problem, as a piece of the current e-learning scenario for students in Pharmacy (similar with that proposed for students in Medicine):

If this dose of digitalis is injected into each of a group of 11 frogs, what is the probability that the number of deaths will be 7 or more?

To answer this question, it is used the same distributional model as for students in Medicine, and the solution is given by the same sum of probabilities. The numerical result is the same, because the binomial template is the same, with the same values for parameters, but with specific texts (see Figure 8).

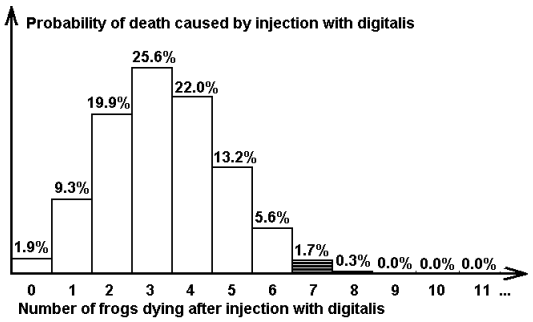

Fig. 8. A graphical solution for students in Pharmacy (dashed columns)

5 The implementation of the bootstrapping e-tools

We implemented *bootstrapping methods* and we created bootstrapping e-tools [Prodan and Prodan, 2002] for simulating laboratory works and experiments. Both the students and the teaching staff use traditional statistical methods to infer the truth from sample data gathered in laboratory experiments. However, the repeated laboratory experiments mean the consumption of a great deal of substances and reactants. At the same time, there are some ethically motivated reasons to reduce the number of animals (guinea pigs, frogs, etc.) used in experimentation. Using a bootstrapping tool and the computer power, the experimenter can repeat the original experiment on computer, obtaining pseudo-data as plausible as those obtained from the original experiment.

Based on distributional models available in JAR (Java ARchive) libraries as infrastructure, we implement in bootstrapping e-tools both parametric and non-parametric bootstrapping methods. When we can not assume the distribution of the population from which the original sample v is taken from, we use the non-parametric bootstrap. When we can make safely assumptions about the distribution of v, we may use the parametric bootstrap. We may use the sample v to calculate a statistic of interest θ^* that is an estimator of some population parameter θ. If we could obtain more samples, we evaluate the estimator on each of these samples. In fact, we only have the one actual sample to work with, so the idea of bootstrapping is to simulate not from the population, but from the single actual sample which we have available.

We have to simulate the population and to generate more so called *bootstrap samples*, or *re-samples*, then to calculate the statistic of interest θ^* for each re-sample, named the *bootstrap replication*. The bootstrapping e-tools provide a set of procedures and functions for re-sampling, for hypothesis testing and for obtaining standard errors, confidence intervals and other measures of uncertainty. The basic bootstrap reassigns randomly the original data and recalculates the estimates. As a computer-intensive method, the bootstrap repeats these reassignments and recalculations thousands of times, treating them as repeated experiments. Using a bootstrapping tool and the computer power, one can repeat the original experiment as many times as necessary to satisfy didactic and research activities.

Generally, we implemented the following algorithm for a bootstrapping e-tool:

i) Read the actual sample $v = (v_1, v_2, \ldots, v_n)$ and evaluate the empirical distribution function F_e.
ii) Simulate N independent bootstrap samples $v^{*1}, v^{*2}, \ldots, v^{*N}$, each containing n data values drawn with replacement from v, based on distribution F_e.
iii) Evaluate the bootstrap replication $\theta^*(k)$ corresponding to each bootstrap sample v^{*k}, for $k = 1, 2, ..., N$.
iv) Estimate the standard error by the sample standard deviation of the N replicates.

The distribution of the statistic of interest θ^* is called *bootstrap distribution*. The bootstrap distribution gives information about the shape, center, and spread of the corresponding population parameter θ.

6 Conclusions and future work

We presented a Java framework for stochastic modelling and simulation, used as an infrastructure to create models and to simulate real world activities, phenomena and processes, particularly in health care, patient monitoring and medical education. As future work, we will combine stochastic modelling with new AI (Artificial Intelligence) paradigms, such as Bayesian inference, intelligent agents and case based reasoning, for simulations and for incorporating intelligent strategies in e-learning scenarios. We will write all simulation and visualization classes in Java and will use the XML (eXtensible Markup Language) format to describe the configurations.

In cooperation with Pharmaceutical Technologies Department of the our university, we have to apply bootstrapping methods in modelling and simulation of some drug design experiments. Real experimental data and simulated pseudo-data refer to some tests made for the characterization of drugs with delayed action, so called retard drugs. This approach is useful, the purpose being to improve real data with simulated valid pseudo-data and to reduce the number of actual tests.

References

[Gorunescu and Prodan, 2001]F. Gorunescu and A. Prodan. *Modelare Stochastica si Simulare*. Microinformatica, Cluj-Napoca, 1st edition, 2001.

[Hesterberg *et al.*, 2003]T. Hesterberg, S. Monaghan, D. S. Moore, A. Clipson, and R. Epstein. *Bootstrap Methods and Permutation Tests*. W. H. Freeman and Company, New York, 1st edition, 2003.

[Prodan and Campean, 2004]A. Prodan and R. Campean. Bootstrapping e-tools for simulating laboratory works. In *ICICTE'2004, International Conference on Information Communication Technologies in Education*, pages 489–494, Samos Island, 2004. University of Athens.

[Prodan and Prodan, 2001]A. Prodan and Rodica Prodan. Stochastic simulation and modelling. In *ETK-NTTS'2001, Exchange of Technology and Know-how - New Techniques and Technologies for Statistics*, pages 461–466, Crete Island, 2001. JRC-ISIS, European Commission.

[Prodan and Prodan, 2002]A. Prodan and R. Prodan. A collection of Java class libraries for stochastic modelling and simulation. In *ICCS'2002, International Conference on Computational Science*, volume 1, pages 1040–1048, Amsterdam, 2002. Springer.

[Prodan and Prodan, 2003]A. Prodan and R. Prodan. A Java framework for intelligent and practical e-learning tools. In *ICICTE'2003, International Conference on Information Communication Technologies in Education*, pages 45–52, Samos Island, 2003. University of Athens.

[Prodan *et al.*, 1999]A. Prodan, F. Gorunescu, and R. Prodan. Simulating and modelling in Java. In *POOSC'99, Workshop on Parallel/High-Performance Object-Oriented Scientific Computing*, pages 55–64. Zentralinstitut für Angewandte Mathematik, 1999.

[Prodan, 2004]A. Prodan. An intelligent and practical educational environment. In *CBLIS'2003, Computer Based Learning in Science*, pages 229–239, Nicosia, 2004. University of Cyprus.

[Ross, 1990]S. M. Ross. *A Course in Simulation*. Maxwell Macmillan, New York, 1st edition, 1990.