Teaching System for Advanced Statistics

Extended Abstract

Ivan Nagy
Faculty of Transportation Sciences CTU
110 00 Prague 1
Prague, Czech Republic
nagy@utia.cz

Jitka Homolová
Faculty of Transportation Sciences CTU
110 00 Prague 1
Prague, Czech Republic
jkratoch@utia.cz

Evgenia Suzdaleva
Institute of Information Theory and Automation
182 08 Prague 8
Prague, Czech Republic
suzdalev@utia.cas.cz

ABSTRACT

A teaching system for presentation mathematical texts in a close connection with software realization of the results is presented. For creating texts, the "pdf-version" of mathematical text editor LaTeX is used. For software application, the interactive programming language GNU Octave is exploited. Both the texts and programs are free, the subroutines used are open. The system is used for teaching the basis of Bayesian Statistics.

Categories and Subject Descriptors

H.7 [**Good practices of integration of FOS solutions into real institutional settings**]: Miscellaneous

General Terms

Teaching or self-teaching based on hypertexts with application of free and open software.

Keywords

Statistics, textbook, free hyperref text, free software subroutines

1. INTRODUCTION

Free learning systems based on the Internet are very popular, nowadays. This system of teaching, either aimed at self-study or to be combined with some regular school study, brings about many advantages. The most evident merits are: (i) the students who missed some lecture or did not understand fully its content have a background where it is possible to fill in the information or to exercise topics that need practical skill; (ii) the students, who prefer to stay at home and to study alone or those who do not study and just want to learn something from the subject, have all they need for their self-study; (iii) the teachers, whose range of lessons is currently cut, become time independent and can rely on the database of information stored on web and so within the reach of everybody.

The situation is complicated if the subject to be taught is of a mathematical nature. Up to now, mathematical texts on web pages are restricted. In our system, a systematic use of "pdf-files" with "hyperrefs" created in the mathematical editor LaTeX proved to be a satisfactory solution.

Another trouble occurs, when the taught subject needs programming and running results on a computer. In this respect, there are two main problems: (i) How to combine web and running a program? (ii) How to describe programs implementing mathematical formulas?

The subject we deal with is STATISTICS. Evidently, it involves both the above problems - complex mathematical description needing hypertexts and running the results in some suitable (free and open) program.

The content of the work further described is a continuation of the work supported by the Minerva EU grant Edukalibre: Libre Software Methods for E-Education [1, 2, 3] and it is based on the theory of Bayesian Modelling Estimation and Control systematically developed in the Institute of Information Theory and Automation, Czech Academy of Sciences [4, 5]. Here we prepared the basic methodology for creating such system and here we also prepared most of the texts and necessary subroutines. This system was preliminary tested on chosen PhD students and now, it started to be used for teaching.

The teaching system mentioned is by no means a unique one. Especially in connection with STATISTICS, there is a lot of such projects. We can divide them into several groups. They are:

1. **Components of encyclopedias**, e.g. Wikipedia, PlanetMath, Encarta, etc. They bring information but they are not systematic.

2. **Professional programs** like Statistica, Statgraphics, Excel, Matlab. They involve relatively wide range of problems and possess high comfort for users. Nevertheless, they are not open, so you can use only offered tasks. Moreover, they are very expensive.

3. **Free programs** as for example First Bayes [6], STEPS [7], Sila [8], Statistical Java [9], MacAnova [10], StatCalc [11], Statistics101 [12], SciLab [18], ViSta64 [13]. There is a lot of these systems, the above choice is

more or less accidental. Mostly they are very useful for practical experiments. The drawback is that they are not directly connected with any teaching text and they are not so easy to extend.

4. **Syllabi of universitas.** They are usually in a standard extend, some go deeper but often at the expense to their range. Some offer the contents, only, some give at disposal also the teaching texts. In any case, they serve to the purposes of the individual schools.

5. **Books on Statistics.** There is a lot of excellent textbooks. However, they usually are separated from programs and they are rather expensive.

6. **Web applications** as can be for example found under the titles Statistical Computing Resources [14], Maths, Stats & OR Network [15] or The Math forum & Drexel [16], which are most close to our demands. They are composed of a continuous presentation of the theory, suitably filled in by applets, demonstrating the specific problems. They give nice demonstration but their modification is not easy. From our point of view, they are practically closed.

The main demands to our teaching system are: (i) connection of systematic exposition of statistical theory with a suitable program; (ii) usage of hyperrefs not only in the text but also in connection with the programs; (iii) the demand that the system is open and free. The presented system is nothing more than an attempt of a union of the demands mentioned.

2. PROBLEM FORMULATION

The goal is to construct a teaching system for training PhD students in the basis of BAYESIAN STATISTICS. As the subject is rather broad and theoretically difficult, only four most important basic tasks have been chosen for presentation. They are

- **Modeling and simulation**, where a description of dynamic system under uncertainty is introduced.
- **Estimation**, which assigns parameter values to a structurally built model so that it would reflect the investigated system.
- **Prediction**, which shows the future behavior of the system.
- **Control**, which computes optimal values of the model controlled variable so that the measured system output would minimize some given criterion.

The teaching system has to

- enable accepting high quality mathematical texts;

- be at disposal at the Internet for easy reach;

- support statistical programs that are

(i) free, (ii) open, (iii) well described.

By "free" we mean to be at disposal for everybody without a necessity to pay for it; "open" denotes a system, where all basic programs are prepared and there is a possibility to add easily new programs based on the existing ones; "well described" guarantees that all mathematical formulas and algorithms contained in the program are clearly described by a mathematical text.

3. THEORETICAL BACKGROUND

The BAYESIAN STATISTICS is a powerfull tool for modelling, estimation, prediction and control of dynamic systems under uncertainty. Here, we sketch only the basis of the named and most important tasks.

First of all, let us introduce a **notation** used. We monitor a process and measure its variables in a discrete time $t = 1, 2, \ldots$. The time instants 0,−1,... denote measurements performed before the beginning of monitoring, which bring a prior information about the process and serve together with measured data for identification. By y_t and u_t we denote the actual (current) values of system output and input. By star, e.g. y_t^*, ve denote all possible values of the variable, here of y_t. The couple of the actually measured data is $d_t = \{y_t, u_t\}$. A set of all measured data up to the actual time instant t (including the prior ones) is denoted $d(t) = \{d_t, d_{t-1}, \ldots\}$. *Note*: Sometimes these sets are interpreted as vectors.

3.1 Modeling

A general description of dynamic system under uncertainty is the following conditional probability density function (PDF)

$$f(y_t|u_t, \phi_{t-1}, \Theta), \tag{1}$$

where

$\phi_{t-1} = \{d_{t-1}, d_{t-2}, \ldots, d_{t-n}\}$ is a regression vector where n is a depth of memory of the model,

θ is a model parameter.

The meaning of such description is following: for given control u_t, regression vector ϕ_{t-1} and parameters Θ the model gives probabilistic description (i.e. PDF) of the output y_t. According to the nature of the random variables entering, the PDF can be either of continuous or discrete nature. Subsequently, the model is called either continuous or discrete.

3.2 Estimation

Bayesian probabilistic description of unknown parameter Θ at time t is the conditional PDF

$$f(\Theta|d(t)) \tag{2}$$

This is so called posterior PDF which includes all the information brought by measured data $d(t)$ (including the prior information) into the parameter description. The PDF $f(\Theta)$ is called prior and it is built only on the prior knowledge.

The proces of collecting information and using it for improving the description (2) in estimation is given by the Bayes rule

$$f(\Theta|d(t)) \propto f(y_t|\phi_{t-1}, \Theta) f(\Theta|d(t-1)), \tag{3}$$

where the principle of Natural Conditions of Control saying that $f(\Theta|u_t, d(t-1)) = f(\Theta|d(t-1))$ (see [4]), is used.

Note: The Bayes rule (3) says that recursive building of information into the parameters description is obtained by repetitive multiplying the parameter PDF by the model PDF. For practical computations, it is very important to choose so called reproducing form of the prior PDF which would preserve its form during the proces of estimation. If not, the computations become infeasible.

3.3 Prediction

The task of one-step-ahead prediction is to forecast the coming output value of the monitored process. Its probabilistic description is again in the form of a conditional PDF

$$f(y_t|u_t, d(t-1)), \tag{4}$$

where knowledge of the parameter Θ is not admitted.

Construction of the predictive PDF (4) using the parameter PDF (2) is following

$$f(y_t|u_t, d(t-1)) = \int_{\Theta^*} f(y_t|u_t, \phi_{t-1}, \Theta) f(\Theta|d(t-1)) d\Theta \tag{5}$$

Note: If we predict more than one step ahead, we talk about a multi-step prediction. This task is more general but also more complicated for computation. Here the output i-steps-ahead is forecasted, based on the knowledge of past data and inputs, or some their model, on the prediction interval, i.e. for time instants t, $t+1$, $\dots$, $t+i$.

Using the predictive PDF (4), the optimal point prediction $\hat{y_t}$ can be computed. For quadratic criterion of optimality, the point prediction is given as the conditional mean

$$\hat{y_t} = E[y_t|u_t, d(t-1)] = \int_{y_t^*} y_t f(y_t|u_t, d(t-1)) dy_t. \tag{6}$$

3.4 Control

The most general description of the feedback control variable has again the form of conditional PDF

$$f(u_t|d(t-1)). \tag{7}$$

Practically, the optimal control variable u_t^* minimizing quadratic criterion $\sum_{t=1}^{N} \omega_t$ with $\omega_t = y_t^2 + qu_t^2$, where q is a penalty imposed on inputs and N is length of control horizon, is deterministic and given by the following formula

$$u_t^* = \arg\min_{u_t} E[\omega_t + \Phi_{t+1}|u_t, d(t-1)] = \tag{8}$$

$$= \arg\min_{u_t} \int_{y_t^*} (y_t^2 + qu_t^2 + \Phi_{t+1}) f(y_t|u_t, d(t-1)) dy_t,$$

where Φ_{t+1} is the partial minimum from the previous step of minimization $\Phi_t = \min_{u_t} E[\omega_t + \Phi_{t+1}|u_t, d(t-1)]$ with $\Phi_{N+1} = 0$ and the needed PDF is the predictive one (4). If the parameter Θ of the model is known, this PDF coincides with the model one (1). If Θ is unknown, the predictive PDF has to be constructed according to (5).

Note: Similarly as for the prediction, the task of optimal control on a finite horizon is more practical and also complicated. Here, the well known principle of dynamical programming must be used. However, such control synthesis is not feasible if an optimal solution is demanded for unknown model parameter Θ - so called dual control. For practical applications, some approximations have to be accepted.

4. TEACHING SYSTEM

The theory sketched in the previous section on a general PDF level can be further elaborated for the whole family of specific distributions of the model. In the teaching system considered, two distributions are considered. The first one is normal, the second one is general discrete distribution. All the theory mentioned is rather sophisticated and the computations following from it are not solvable without using a computer. From the facts mentioned above, it follows:

- a thorough and well structured theory description with hyperrefs is highly desirable,
- free and open software implementing the algorithms following from the theory is necessary,
- as close as possible connection between the theory and the programs is essential.

The teaching system should respect all these demands.

4.1 Hypertexts

As it has been mentioned, understanding of such a complex subject as BAYESIAN STATISTICS is impossible without intelligible and well organized text. Not only does the text have to be written carefully but it also has to include automatic references. They provide the reader quick approach to referenced equations or explanation of some basic notions. They also can evoke the subroutines implementing the algorithms and thus they connect the theory and the corresponding program.

There are two main types of hyperrefs. Firstly, they are ordinary references to equations or sections fully supported and used practically in all text editors supporting mathematics. Secondly, they are references to a certain database. There are approximately three types of databases used in the teaching system.

Database of notions

To understand the theory correctly, it is very important to remember well the precise meaning of all defined notions. It is obviously difficult, especially for students. To help them, all the basic notions are made as hyperrefs and after clicking they show a brief description of the corresponding notion from the prepared database. The database of notions contains also inter-references to other terms of the database and the whole database is equipped with index. It enables the students to use the database itself also as a teaching text just through surfing it and looking for meaning of notions and their connections.

Database of subroutines

The basic statistical algorithms are supported by subroutines. They can be trivial (e.g. for model simulation) but also rather complex (e.g. for control synthesis on a finite interval). In any case, they need a good and thorough description. As the subroutines implement mathematical algorithms, they need also mathematical description. Internal program comments are totaly insufficient. A good solution is to create a database of subroutine descriptions, each description connected to its subroutine by a hyperref. Naturally, the hyperrefs are written also across the database thus enabling to browse the database and to learn the links between the subroutines.

Database of tasks
The theory results into specific problems that are called tasks. The implementation of these tasks can be built from the selected subroutines. E.g. a task of prediction the output of a dynamic system can use subroutines for simulation, statistics collection, computation of parameter point estimates and prediction. Similarly as for subroutines, these tasks deserve good mathematical introduction that is done through the database of the tasks descriptions. Both the databases (for subroutines and tasks) are very important parts of the theoretical text.

An ideal solution for writing such texts is to use the free mathematical editor LaTeX (in its "pdf-variant") which it is able to generate mathematical texts of high professional quality. After using the package "hyperref" the texts are supplied not only by automatic references to equations and sections, but it is also possible to make named anchors in the text and to produce automatic references to them as well as to other "pdf-files" and anchors in them.

4.2 Software

There is a lot of statistical software at disposal with various quality in communication and presentation of results. Some of them are really excellent. But, as far the authors know, they all have the same common drawbacks: they are not open and they are rather expensive.

Openness of the software
By openness we mean a possibility of easy programming new subroutines and using them for a construction of new tasks. If this feature is missing, nomatter how broad the basis of the subroutines is, a student can always come to its border and this is crucial to him. He is restricted and cannot solve his problem that is out of range of program possibilities.

That is why, it is necessary to use some programmable language as a statistical software. In this software, a basic collection of subroutines as well as tasks must be prepared with a sufficient luxury of setting the tasks and at least average quality of both numerical and graphical outcomes. These subroutines and tasks must be supplied in a source code to (i) provide a student information about its content, (ii) be a guide how to program further subroutines or tasks.

It is clear, that programming in the chosen language must be as simple as possible not to task student's mind by other thing than STATISTICS. From this point of view, the interactive program language GNU Octave seems to be an ideal choice. It is not only simple and powerfull but also free of charge.

Price of the software
A crucial problem for students, studying STATISTICS, is that they are instructed in some statistical program which is hired to them for exercising during the seminars. A great deal of work is done by a student to master the program. Even an examination often tests the level of mastering the programm. But after finishing the course of STATISTICS, the program becomes inaccessible to a student because it is too expensive to buy. That is why a big stress is put on the fact, that the software used for teaching STATISTICS is free.

4.3 Connection between theory and software

STATISTICS is the subject where neither theory nor practical application can be suppressed. If only the theory is stressed, the students can memorize formulas and propositions, but they are not able to solve practical tasks. That is why the approach of many statistical courses is to teach software, first of all. Nevertheless, this approach can be even more dangerous, because students learn only clicking the mouse on the buttons of the program. They learn just to solve several standard situations in the program and very often they are even not able to interpret the result.

Good connection between theory and software is necessary from the following reasons

- the text of a subroutine can be a good help in understanding the corresponding theoretical algorithm,
- the theoretical text is necessary for a correct choice of the proper subroutine,
- the theoretical description of subroutines is unreplaceable help for their correct setting as well as interpreting the obtained results.

An idea of solution the problem is following:

At the top level there is the theoretical text, dealing with the four general problems on the level of pdfs without any specification to a particular model distribution. The problems are modeling, estimation, prediction and control. Each problem has pointers to the specific tasks.

Under the topmost level there is a level of tasks. Here the general problems are specified for a particular model distribution. Here the mathematical specification of the task as well as its program implementation are described. An important component of this file is also a reference to the program implementation of the task as well as all the related subroutines or tasks. The collection of all task descriptions creates a database of the tasks, represented by its index.

The lower supporting level is a collection of subroutines, used for a construction of the tasks. The philosophy of their description is the same as for the tasks. Each subroutine has its "pdf-file" with a description. This description is on one hand connected to the tasks using the particular subroutine on the other hand and above all it is bound to the subroutine itself. A collection of all subroutine descriptions creates a database of subroutines, again with its index.

Besides this "pdf-description", each task and subroutine has its own comments as already mentioned above.

5. EXAMPLE

Let us demonstrate the teaching system on some simple example. Consider a static discrete system with the output $y \in \{1, 2, \dots, n\}$. Its model is parametrized by the vector parameter $\Theta = \{\Theta_1, \Theta_2, \dots, \Theta_n\}$. The task is to perform a point estimation of items of the model parameter (in the following text, the underline means hyperref).

5.1 Theory

Model

In the theoretical description, the model (1) specifies to

$$f(y_t|\Theta) = \prod_{i=1}^{n} \Theta_i^{\delta(i,y_t)}, \tag{9}$$

where $\delta(i, y_t)$ is the Kronecker function which is equal to one for $y_t = i$ and is zero otherwise and $\Theta_i \geq 0$ for $i = 1, 2, \dots, n$, and $\sum_{i=1}^{n} \Theta_i = 1$.

This model directly expresses probabilities that the system will be in one from his n states determined by the specific value of his output.

Estimation

The general Bayes rule (3) gets the form

$$f(\Theta|d(t)) \propto f(y_t|\Theta)f(\Theta|d(t-1)) \qquad (10)$$

with the model PDF (9) and the prior PDF in the self reproducing form

$$f(\Theta|d(t-1)) = \prod_{i=1}^{n} \Theta^{V_{i;t-1}-1}, \qquad (11)$$

where V_{t-1} is a vector statistics at time $t-1$. The statistics V_0 is the prior statistics expressing an expert knowledge.

Inserting (11) and (9) into (10) we obtain the rule of updating the statistics by sequentially measured data

$$V_{i;t} = V_{i;t-1} + \delta(i, y_t), \qquad (12)$$

for $i = 1, 2, \ldots, n$ and on the whole time interval for which the estimation is to be performed.

Note: The meaning of the previous update is clear and it fully agrees with the statistical definition of probability. Each time a new data item comes we find to which state (level of output) it belongs and we increment the corresponding item of the statistics by one.

After updating the statistics we can compute the point estimates of the parameter items. The most widely used choice for optimal selection of these point estimates is to minimize the following quadratic criterion

$$E[(\Theta - \hat{\Theta}_t)' Q (\Theta - \hat{\Theta}_t)|d(t)],$$

where $\hat{\Theta}_t$ is the point estimate involving data up to time t and Q is some penalization matrix.

Minimization of this criterion gives the point estimate in the form of conditional expectation

$$\hat{\Theta}_t = E[\Theta|d(t)]. \qquad (13)$$

Prove as a homework or see point estimates.

This result applied to our example gives

$$\hat{\Theta}_{i;t} = \frac{V_{i;t}}{\sum_{j=1}^{n} V_{j;t}}, \qquad (14)$$

where V_t is the vector statistics from (11). Proof: see here.

Database of notions

Here is an example of the database of notions mentioned. Some of the notions involved in the theory example are used here for illustration.

Point estimate is a number which estimates the value of the unknown parameter. The formula for computation of an optimal point estimate depends on the choice of optimality criterion. E.g. for quadratic criterion the point estimate is a conditional expectation conditioned by all measured data.

Quadratic criterion at time t for variable X and reference value $\hat{X}_t$ is

$$E[(X - \hat{X}_t)' Q (X - \hat{X}_t)|d(t)],$$

where Q is a weighting matrix - for no weighting it is a unit matrix.

Statistics is a (vector) variable that comprises all the collected information necessary for parameters estimation. By the information we mean the prior (expert) information and that carried by the collected data.

Weighting matrix is matrix entering the quadratic criterion. The magnitude of its diagonal items can stress suppress significance of some items of the penalized variable. The nondiagonal items can even penalize products of different items of the variable - e.g. the increments of the output variable.

Database of subroutines

Here, a sample of the database of subroutines is presented. Each item of the database has two parts. Firstly, it is a descriptive part, written in a "pdf-form", describing the subroutine mainly from theoretical point of view. Secondly, it is the subroutine itself with the comments inside it, describing mainly the meaning of input and output parameters.

Subroutine simDisc

The subroutine performs simulation with a discrete static model of the form

$$f(y_t|u_t, \varphi_t\Theta) = f(y_t|\Theta) = \Theta_{y_t}.$$

This model chooses values of $y_t = i$ from a set {1,2,...,n} so that $Pr(y_t = i) = \Theta_i$.

Program implementation is following:

`Th=cumsum(th);`

performs cumulative summation e.g. for `th=[.3 .2 .5]` it gives `Th=[.3 .5 1]`.
Output generation is done by the command

`y=sum(Th<rand)+1;`

where `rand` is a generator of uniformly distributed random variable and `Th` is a cumulative sum of the discrete model parameter.

Example: Let `th=[.3 .2 .5]` then `Th=[.3 .5 1]`. Now, let `rand=.392`. Then `rand<Th=[1 0 0]`. And `y=2`, what is correct as `rand>.3`, and `rand<.3+.2` and `rand<.3+.2+.5`.

Call of the subroutine: `y=simDisc(th)`

Show the subroutine in the editor: push here

Similar subroutines: `simDiscDyna`, `simCont`, `simContDyna`

and here is the corresponding subroutine

```
function y=simDisc(th)
% y=simDisc(th)
% simulation of discrete system
%
% y  system output
%    = 1,2,...,length(th)
% th system parametr
%    = prob. of items of y
%
Th=cumsum(th);     % cummulative probs
y=sum(Th<rand)+1;  % output generation
```

Subroutine statDisUpdt

The subroutine performs updating of a discrete statistics

$$V_{i;t} = V_{i;t-1} + \delta(i, y_t)$$

for $i = 1, 2, \ldots, n$ where n is length of the statistic V.

This update follows from the Bayes rule when substituting the static discrete model and corresponding (self-reproducing prior PDF).

Call of the subroutine: `V=statDisUpdt(V,y)`

Show the subroutine in the editor: push here

Similar subroutines: `statConUpdt`

and here is the corresponding subroutine

```
function V=statDisUpdt(V,y)
% V=statDisUpdt(V,y)
% discrete statistics updata
%
% V  statistics
% y  level of measured output
%     = 1,2,...,length(V)
%
V(y)=V(y)+1;        % statistics update
```

Subroutine ptestDisc

The subroutine performs computing of parameter point estimates based on the estimated parameters PDF (posterior PDF). The computation of the point estimates for quadratic criterion of optimality is following

$$\hat{\Theta}_t = E[\Theta|d(t)] = \int_0^1 \Theta f(\Theta|d(\Theta|d(t)))d\Theta =$$

$$= \frac{B(V_{t-1} + \delta(i, y_t))}{B(V_{t-1})} = \frac{V_{y_t}}{\sum_{j=1}^n V_j}.$$

The following formulas have been used: beta function, multivatiate beta function.

Call of the subroutine: `th=ptestDisc(V)`

Show the subroutine in the editor: push here

Similar subroutines: `ptestCont`

and here is the corresponding subroutine

```
function th=ptestDisc(V)
% th=ptestDisc(V)
% dicrete quadratic point estimates
%
% th  point estimates
% V   statistics
%
th=V/sum(V);        % point estimates
```

Database of tasks

Similarly to the database of subroutines, each item of the database of tasks consists of the similar two parts, too. They are the descriptive part and the executive one. The descriptive part provides complete description of the task. Mainly, it uses a reference to the general theory and then describes a specialization to the specific task.

Then, some difficult programming steps are discussed and explained.

After it, a serial of experiments is recommended. Each experiment is accompanied by a comment, saying what remarkable can be seen or what changes can occur when some parameter is varied.

Conclusion summarizes the meaning of the task and points out other similar tasks.

Here is an example of the task prepared throughout this Example - point estimation of parameters of discrete model. (Again, the underlined text means hyperref.)

Task EstPtDisc

Theoretical solution

The program solves the task of point estimation of parameters of a static discrete model. The general form of a model is given in (1). Its discrete variant is specialized in (9). If as a prior PDF, the self reproducing prior (11) is chosen then the statistics update runs according to (12). The point estimates for quadratic criterion of optimality are computed using the formula (14).

Software solution

The task is constructed from three subroutines:

simDisc that performers simulation of discrete static system,

statDisUpdt that collects statistic from th incoming data and based on a specified prior knowledge,

ptestDisc that computes point estimates for the quadratic criterion of optimality.

Recommended experiments

1. Set various parameters of the simulated system.

   Those items of the parameter that have small probability of occurrence are more difficult to estimate. The reason is a lack of evidence for them.

2. Try various prior probability for estimation. You can do it by changing the coefficient `npri` in the range from 1 to 100.

   For small `npri` the prior (uniform) information is suppressed. The bigger value of `npri` starts with the uniform prior information which calms the start, but brings a slight inaccuracy into the estimation.

Conclusions

The task of prediction is very important from both theoretical and practical point of view. Theoretically it shows how to construct the predictive PDF, that is further used for all tasks of control. These tasks are very complicated and we often have to restrict ourselves to some approximation, where instead of the predictive PDF the point estimates of the model parameters are used. In applications, even the task od prediction itself plays a significant role. For example in transportation problems, it is very important to know the future evolution of transportation flows in the monitored traffic region.

and here is the corresponding task

```
clc,clear all
% Task: Parameter estimation of static discrete
% -----------------------------------------------
% model f(y(t)|th) = th_y(t), y(t)=1,2,...,n
%
% Data for setting
ndat=2000;                     % number of data
npri=10;                       % number of prior d
th=[.1 .4 .3 .2];              % model parameters

% Data for initialization
V=ones(size(th))*npri;
The=[];

% Time loop
for i=1:ndat
  y(i)=simDisc(th);            % simulation
  V=statDisUpdt(V,y(i));       % statistics update
  The=[The; ptestDisc(V)];     % point estimates
end

% Results - prints
disp('Original parameters')
th
disp('Parameter estimates')
the=ptestDisc(V)

% Results - plots
fig,plot(The)
hold on
s=fix(2*ndat/3):ndat;
plot(s,th(1)*ones(size(s)),s,th(2)*ones(size(s))
  s,th(3)*ones(size(s)),s,th(4)*ones(size(s)))
hold off
st1='Evolution of parameters estimates ';
st2='(stight lines at the end are tru values)'
title([st1,st2])
xlabel('Time')
ylabel('Estimates')
axis([1,ndat,min(th)-.05,max(th)+.05])
```

Illustration of results

Results of the tasks can be both numerical and graphical. Here we demonstrate a graphical one which illustrates evolution of point estimates of the parameters of the investigated static discrete model. In the two figures the influence of different prior information is demonstrated.

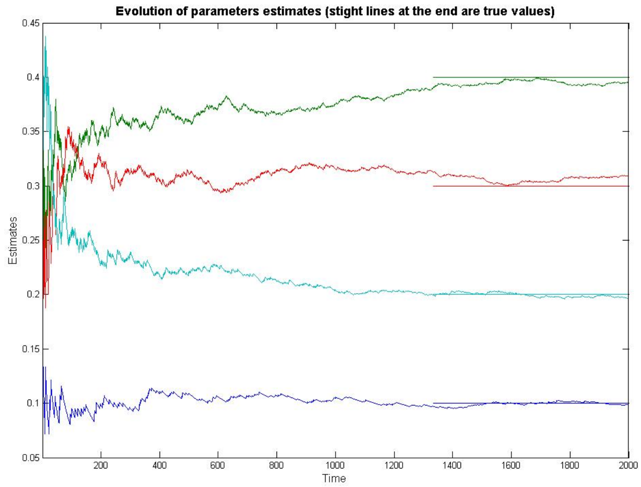

Figure 1: Evolution of point estimates with week prior

In this figure, a sequential refinement of the values of point estimated during the parameter estimation of a static discrete model with four output levels is shown. The prior information given the estimation at the very beginning of the process is uniform distribution (all parameters have the same values) and this information is rather week (as it is gained just from four measurements). That is why the beginning of estimation is rather chaotic.

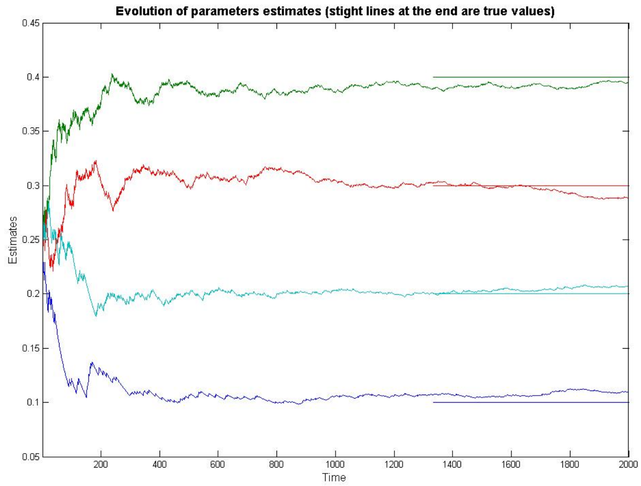

Figure 2: Evolution of point estimates with stronger prior

Here, similar situation is presented. The prior distribution is uniform again, but its weight is as if it would be extracted from 40 data items. Thus the information gained from the initial really measured data is compared with the relatively strong prior information and thus, the jumps given by the randomness of the initial data are reduced.

6. CONCLUSIONS

The teaching system described grows from our demands as lecturers of Statistics. The topic presented concerns rather complicated mathematical area of Bayesian statistics focused to PhD students. As the final target of the course is a practical application, the subject is closely bound with programming. This is what makes the presented subject characteristic.

This year, we started to use the system for teaching on our Technical university, Prague, in the Czech Republic. Even this one-year experience with the system has shown some of its imperfections. The most serious of them is that we must pay much higher attention to examples from the practice throughout the whole explanation. E.g. we should speak not generally about a measured variable but specifically for instance about an occupancy of the traffic flow measured on a detector in a specific point of communication or not about a model needed for prediction but about a model that could predict the traffic flow intensity.

From the fact, that we are the teachers, it follows

- we have permanent audience (each year about 20 persons) that will use the system;
- we will have a stable feedback from this people indicating what is wrong or at least not so easy to understand;
- we will have to work continuously on improving the system as, naturally, each teacher wants to teach as best as possible.

Only years of practice with using the system can acknowledge our hope in its usefulness and to focus it directly on the problems our PhD students are going to meet.

Nevertheless, the significance of the mentioned system is not only for teaching Statistics. It shows new methodology of constructing such a system by those who are not programmers by their profession. It is easily applicable for any other subject with similar characteristics.

7. REFERENCES

[1] *Edukalibre: Libre Software Methods for E-Education*, SOCRATES Programme, Minerva - Promotio of Open and Diastance Learning - Information and Communication Technologies in the Field of Education., 110330-CP-1-2003-1-ES-MINERVA-M.

[2] E. Suzdaleva, I. Nagy, L. Pavelková, and Jitka Homolová, “Edukalibre. guide”, Tech. Rep. 2145, ÚTIA AV ČR, Prague, October 2005.

[3] E. Suzdaleva and I. Nagy, “Bayesian decision making and e-learning: Interactive examples”, Tech. Rep. 2133, ÚTIA AV ČR, Prague, July 2005.

[4] V. Peterka, “Bayesian approach to system identification”, in *Trends and Progress in System Identification*, P. Eykhoff, Ed., pp. 239–304. Pergamon Press, Oxford, 1981.

[5] M. Kárný, J. Böhm, T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař, *Optimized Bayesian Dynamic Advising: Theory and Algorithms*, Springer, London, 2005.

[6] First Bayes, *http://www.tonyohagan.co.uk/1b/.*

[7] STEPS, *http://www.stats.gla.ac.uk/steps/.*

[8] Sila, *http://www.eco.rug.nl/medewerk/knypstra/sila.html.*

[9] Statistical Java, *http://www.stat.vt.edu/ sundar/java/applets/.*

[10] MacAnova, *http://www.stat.umn.edu/macanova/.*

[11] StatCalc, *http://www.ucs.louisiana.edu/ kxk4695/StatCalc.htm.*

[12] Statistica101, *http://www.programplaz.com/company/statistics101-net.html.*

[13] ViSta64, *http://blogs.zdnet.com/microsoftvistas/?p=32.*

[14] Statistical Computing Resources, *http://www.ats.ucla.edu/STAT/.*

[15] Stats & OR Network Maths, *http://mathstore.ac.uk/.*

[16] The Math forum & Drexel, *http://mathforum.org/.*

[17] J.M. Bernardo and A.F.M. Smith, *Bayesian Theory*, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore, 1997, 2nd edition.

[18] SciLab, *http://www.scilab.org/.*