Educational Tools for Dynamic Bayesian Decision-making

EVGENIA SUZDALEVA
Department of Adaptive Systems
ÚTIA AV ČR
Pod vodárenskou věží 4, 18208 Prague
CZECH REPUBLIC
suzdalev@utia.cas.cz

PETR NEDOMA
Department of Adaptive Systems
ÚTIA AV ČR
Pod vodárenskou věží 4, 18208 Prague
CZECH REPUBLIC
nedoma@utia.cas.cz

Abstract: The paper deals with the interactive educational tools, developed for the area of Bayesian decision-making and based on PDFLATEX and $MATLAB^{®}$. The extensive educational system enables to familiarize with the tasks of Bayesian decision-making, related to the probability mixture models, and to learn the theoretical and algorithmic aspects of their solution. The system is intended for the postgraduate students and researchers, beginning the activity in this area. The paper summarizes the principles, used in design of the system, describes the teaching session and demonstrates the exploitation of the system with the help of several examples, related to different Bayesian decision-making tasks.

Key–Words: Educational system, Bayesian decision-making, probabilistic mixtures

1 Introduction

The paper describes the principles of creating the educational means for dynamic Bayesian decision-making. Bayesian decision-making is a well-elaborated statistical methodology, applied in practice for solving variety of problems, dealing with the data mining.

The need for educational system has arisen, since the postgraduate students and the beginning researchers were faced with difficulties in form of tremendous variety and volume of scientific materials to be studied. Then, the theoretical and algorithmic basis of dynamic Bayesian decision-making has been gathered in the book [1], and the interactive educational system, based on this book, has been developed. Software support of the system has been provided by the means of PDFLATEX, $MATLAB^{®}$ [2], and especially by the toolbox Mixtools [3], which is worth mentioning separately. Mixtools is the extensive software toolbox, offering implementation of majority of the decision-making algorithms, presented in [1], and finding successful application in many areas (advisory system for operators of rolling mills [4], diagnostics of thyroid cancer [5], urban traffic control [6], diagnostics and therapy of upper limb lymphedema [7] etc.). Thus, using the algorithms of Mixtools, the educational system covers both theoretical and programming aspects of dynamic Bayesian decision-making. The basic facts of this methodology are briefly described in Section 2.

The main emphasis of the paper is on developing and design of the educational system, that is provided in Section 3. Under developing the system a number of principles was followed. One of them is *connectedness to theory*. The paper describes, how the access to theory is organized. Another important principle is the *use of database*, explaining the terms and meanings of variables. One of the basic features of the system is availability of educational tools both *in PDF format and by means of* $MATLAB$. The paper describes a structure of these tools, which explain the decision-making tasks, and proposes the original way of their specification. Navigation inside the system allows to run the task in $MATLAB$ from the PDF format document, view a $MATLAB$ code, open the corresponding theory, read about the recommended experiments etc. Section 3 describes in details the use and the functioning of the system during the teaching session.

Section 4 demonstrates the exploitation of the educational system. Three educational examples, chosen for different decision-making tasks, are shown. The remarks in Section 5 close the paper.

2 Basic Facts of Bayesian Decision-making

To begin with, Bayesian decision-making requires to introduce a concept of *decision maker*. In the context of the educational system this concept means an user, working with the system, or a student. The next concept to be introduced is a *system*, which the decision maker wants to describe or influence. The decision maker possesses some knowledge about the system – *experience*. It is clear, that there exists some knowledge about the system, which is not available to the decision maker. It is denoted by a concept *ignorance*. The decision maker has a set of *admissible strategies*, which impose the physical restrictions on his actions. Naturally, there is a *loss function*, which quantifies the degree of reaching the decision maker's aim. Under all these conditions the decision maker should make *decision*, which would lead to the reaching of aim. In these words a scheme of decision-making can be briefly summarized.

To describe a system, the decision maker chooses a model. The probability *mixture models* proved themselves as a effective mean for solution of Bayesian decision-making tasks. The mixture model is the conditional probability density function

$$f(data \mid previous\ data,\ parameters),$$

which is presented as the sum of the weighed components

$$\sum_{c=1}^{ncom} weights_c \times \tag{1}$$

$$\underbrace{\prod_{k=1}^{K} \underbrace{f_k(data \mid previous\ data,\ parameters).}_{k\text{th } \textit{parameterized factor} \text{ within } c\text{th component}}}_{c\text{th } \textit{component}}$$

Let's suppose, that the decision maker wants to describe a system by the mixture. The system data, which should be preprocessed, are available to the decision maker. Then, some prior knowledge about the system might be available. For example, it can be the knowledge whether the system is a static or the dynamic one. This is the *experience* of the decision maker for this moment of time.

For a start, the decision maker wishes to estimate a mixture structure. In other words, he has to perform the *mixture initialization*, which represents a construction of prior estimate of the mixture. The result of initialization gives the decision maker a mixture model structure.

After initialization it is necessary to perform the *mixture estimation*, that results in a better quality mixture with estimated parameters.

The mixture model obtained should be verified before its using in subsequent tasks. In other words, it is necessary to check whether the model provides good description of system data or not. This task is called *model validation*.

The tasks mentioned, together with their variations, belong to one of the basic branches of Bayesian decision-making – *learning*, here learning with mixtures. This is a schematic sequence of actions of the decision maker, when the aim is only to *describe* the system. Naturally, this global aim can include many various sub-aims.

In the case, when the aim is to *influence* or *control* the system, the task of the decision maker belongs to another basic branch of Bayesian decision-making – *design*. It contains several types of *control design* and *design validation*.

All these tasks have been described here very briefly, only to give a general idea of decision-making process. To solve each of them, the variety of techniques and algorithms are used. Most of the algorithms have been implemented in toolbox Mixtools, mentioned above. The problem has arisen, how to teach the students and young researchers the theory and algorithms of Bayesian decision-making.

3 Design of the Educational System

The educational system is organized in the form of collection of the interactive case studies, each of which explains a decision-making task, depending on system data, model used, solution technique etc. The case studies are self-contained. They provide information enough for getting acquainted and understanding the decision-making tasks. During the teaching session the educational system invites a student to play a role of *external decision maker*, while the task is solved internally by the case study. Under developing the educational system the following principles were conformed.

Connectedness with theory The system is connected with a theoretical background both in a direct and a figurative sense. The text of the case studies, the concepts and variables are based on the book [1] and correspond to the terminology, introduced in it. The case studies provide the hyperlinks to the articles, or to the educational text with theoretical material, depending on the decision-making area, being discussed. For example, the case study on mixture estimation with one of the estimation algorithms allows to open the papers about algorithms and the mixture estimation in general. There is no more need to look for the theoretical materials in the library.

Use of the database The system is connected with a decision-making database. It contains definitions of the decision-making terms and explains the meaning of the variables. All the decision-making concepts (for example, *system*, *experience*, *mixture* etc) and the variables, met in the text, are the hyperlinks to the labels of database. Clicking on them shows the definition of the concept in new window.

Use of PDFLATEX and $MATLAB$ The educational system was developed in PDFLATEX, which offers the high-quality tools not only for typing the scientific documents, but also for using the hypertext references inside the system. Moreover, the hyperlinks in such a PDF format document allow to run the task in $MATLAB$. The educational system exploits $MATLAB$ to run the tasks, but any similar software can be used. For example, the similar educational system [8] used GNU Octave [9]. In this sense the system has property of *universality*. Moreover, most of the processing functions of the toolbox Mixtools are re-coded in C-language and used as MEX-functions.

Thus, the educational system unifies the text description of the decision-making task and its program implementation. It means, each case study is offered as a text in the PDF format and as $MATLAB$ script.

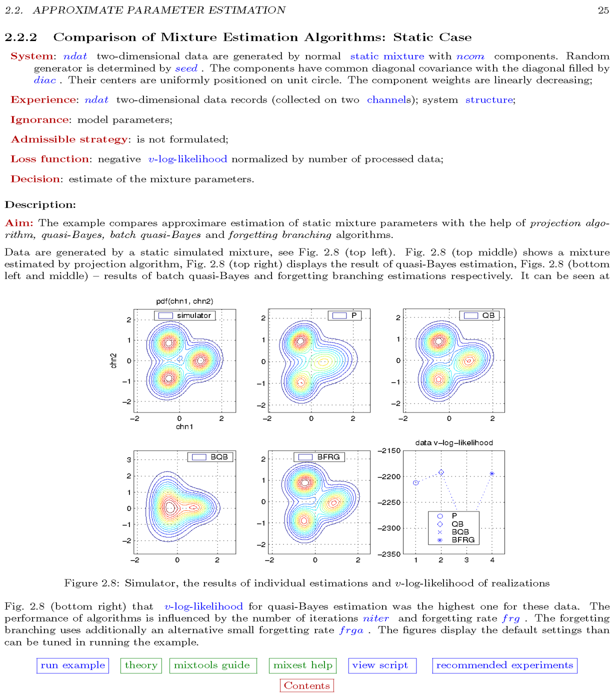
Case study in the PDF format Each case study has its own page in PDF format. Example of such a page is shown at Fig. 1.

At the beginning the page contains the *specification* of the decision-making task. It consists of the concepts defined in Section 2:

- ○ *system*,
- ○ *experience*,
- ○ *ignorance*,
- ○ *admissible strategy*,
- ○ *loss function*,
- ○ *decision*.

The specification allows the decision maker to have a clear idea of the task to be solved. With its help, the decision maker understands what means (*experience*, *admissible strategies*, *loss function*) are available to him for reaching the aim. This way of specification of the decision-making task was introduced in [1].

Then, the page offers the solution of the task. The solution usually involves the algorithms of toolbox Mixtools. It is described from the position of performance of the algorithms with respect to the concrete task. The description of the solution is illustrated by the figures, where primarily demonstrate the data, results of the individual iterations and the final result of the solution (for example, estimated model). The description of the solution interprets the results and gives an insight into performance of algorithms.


2.2. APPROXIMATE PARAMETER ESTIMATION 25

2.2.2 Comparison of Mixture Estimation Algorithms: Static Case

System: *ndat* two-dimensional data are generated by normal static mixture with *ncom* components. Random generator is determined by *seed* . The components have common diagonal covariance with the diagonal filled by *diac* . Their centers are uniformly positioned on unit circle. The component weights are linearly decreasing;

Experience: *ndat* two-dimensional data records (collected on two channels); system structure;

Ignorance: model parameters;

Admissible strategy: is not formulated;

Loss function: negative v-log-likelihood normalized by number of processed data;

Decision: estimate of the mixture parameters.

Description:

Aim: The example compares approximare estimation of static mixture parameters with the help of *projection algorithm, quasi-Bayes, batch quasi-Bayes* and *forgetting branching* algorithms.

Data are generated by a static simulated mixture, see Fig. 2.8 (top left). Fig. 2.8 (top middle) shows a mixture estimated by projection algorithm, Fig. 2.8 (top right) displays the result of quasi-Bayes estimation, Figs. 2.8 (bottom left and middle) – results of batch quasi-Bayes and forgetting branching estimations respectively. It can be seen at

Figure 2.8: Simulator, the results of individual estimations and v-log-likelihood of realizations

Fig. 2.8 (bottom right) that v-log-likelihood for quasi-Bayes estimation was the highest one for these data. The performance of algorithms is influenced by the number of iterations *niter* and forgetting rate *frg* . The forgetting branching uses additionally an alternative small forgetting rate *frga* . The figures display the default settings than can be tuned in running the example.

Figure 1: Example of a case study page

The important terms and the variables, met at the PDF page, are the color hypertext references to the database of definitions.

At the bottom of the page there are several navigation buttons. They are worth describing in details. The button “theory” refers to the articles depending on the decision-making area discussed. The purpose of the button “mixtools guide” is clear – it opens the manual [10] to the toolbox Mixtools at the chapter, related to the discussed area. The next button is “function help”. It gives a description of the function from the toolbox Mixtools, which has been used for solving the decision-making task. The button “recommended experiments” informs the decision maker, the changes of what values can influence on the performance of the algorithms. It gives a list of experiments for the decision maker to fulfill. The button “mainpage” returns the decision maker to the main page of the educational system. The mentioned buttons always open the PDF document. It means, that the articles, the database, guide and description of functions are of PDF format.

The rest of the buttons – “run example” and “view script” – are of another nature. The button “view script” shows a $MATLAB$ script of the case study in some simple text editor. The button “run example” starts the task in $MATLAB$.

Case study in $MATLAB$ The case study in $MATLAB$ represents a script, which uses the decision-making algorithms of the toolbox Mixtools. This script produces the $MATLAB$ dialog. The dialog shows the elements of the specification of the decision-making task already with certain values of the variables and settings of the algorithm (number of iterations, options etc). The dialog asks whether the decision maker wants to change something or not. Then the task is solved. The results of the solution are plotted at the figures, which have been shown at the PDF format page of the case study. It should be noted, that the PDF format page always contains the results with default settings of the algorithm. Then $MATLAB$ shows the dialog with the changed values and proposes to continue the solving, or to return to the default values, or to stop.

Teaching session Thus, the educational system requires any PDF viewer and $MATLAB$ installed on the computer. The teaching session includes the following actions of the decision maker. The decision maker opens the PDF format document and gets into the system main page. It is necessary to choose the case study with task of decision maker’s interest and to read the PDF format page. In the case of need the decision maker uses the hyperlinks to the database of terms, to the theoretical materials and Mixtools guide. Then, with the help of button “run example”, the decision maker runs the task in $MATLAB$. $MATLAB$ presents the solution with the default settings. After getting acquainted with it, the decision maker returns to the case study in PDF format and obtains the list of experiments to be done with the help of the button “recommended experiments”. He switches into $MATLAB$, makes the experiments and observes the influence of the changed values on the performance of the algorithm. More advanced decision maker can learn the $MATLAB$ code of the case study, which is demonstrated by means of the button “view script”. Having learned the task, the decision maker returns to the main page of the system and chooses another one.

4 Exploitation of the Educational System

To demonstrate the use of the system, three case studies have been chosen. The first one concerns the task of the *mixture initialization*, see Section 2.

4.1 Example 1

The title of the case study at the main page of the systems sounds: *Static Mixture Initialization – “Banana Shape” Benchmark*. The aim of the example is to show the result of initialization for one of the most easily visually verified form of mixture.

The PDF format page of the case study contains the following information (note, that all the terms typed by italics are the color hyperlinks to the database of definitions).

System: *ndat* data are generated by the static *normal mixture* with *ncomi* components. The components are located in the shape of banana, which can be easily used for the visual model verification. Random generator is defined by *seed*. The common covariance of components is *diac*;

Experience: *ndat* data; knowledge that the system is a static one;

Ignorance: mixture structure; mixture parameters; initial setting of the initialization algorithm;

Admissible strategy: formulated only for design tasks, here it is not formulated;

Loss function: negative v*-log-likelihood*;

Decision: estimation of mixture *structure* and parameters.

The banana-shaped mixture is taken as the simulator, see Fig. 2 (top left). The simulator generates the data. The order of data is randomized to eliminate possible influence of the mixture-based data generation. The data, scaled to zero mean and unit variance, are displayed at Fig. 2 (top right). Then the initialization of mixture is done, using the Mixtools function *mixinit*, see Fig. 2 (bottom left). The performance of the initialization algorithm is influenced by a number of components of the initial model *ncomi* and a number of iterations *niter*. The settings of the algorithm are defined by the variable *opt*.

The initialized mixture should be checked whether it provides a good description of data or not. Due to its shape this model can be verified visually. The initialized mixture is used as the simulator and generates the data of the same size, see Fig. 2 (bottom right). The good visual correspondence of the data, used for initialization, and the data, generated by the initialized mixture, validates the model. Fig. 2 demonstrates the results with default settings of the algorithm, that can be changed in running the case study in $MATLAB$.

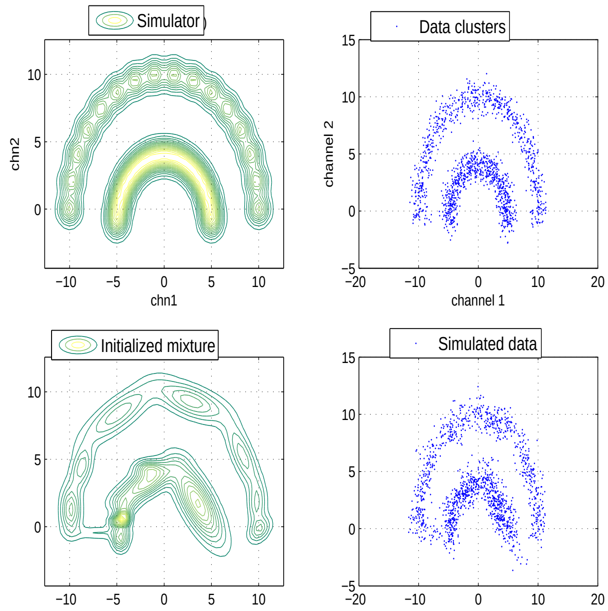


Figure 2: Simulator, scaled data clusters, the initialized mixture and data, generated by it

Navigation button “function help” gives the description of Mixtools function *mixinit* and its parameters. The button “run example” starts $MATLAB$, which produces the following dialog:

```
--- System ----------------------------
number of components       ncom  = 40;
diagonal of noise cov.     diac  = 0.5;

--- Data -----------------------------
length of data             ndat  = 1500;
seed                       seed  = 1234;

--- Initialization -------------------
prior No. of components    ncomi = 10;
number of repetitions      niter = 3;
options          opt = ’pn1k3g3h100d100’

any changes?
ENTER = no, or type a command: >
```

The default values of variables are shown. Note the variable *opt*, description of which the decision maker obtains in the database. The variable *opt* includes several options (here *p*, *n1*, *k3*, *g3*, *h100*, *d100*), which indicate the settings of the initialization algorithm, in one default value.

The decision maker can change the values of variables according to recommendations, obtained in PDF format page by means of the button “recommended experiments”. It is recommended to observe the performance of the algorithm with the following values:

- ○ number of mixture components *ncom* = 10, 20;
- ○ diagonal of noise covariance *diac* = 0.1, 0.5;
- ○ data size *ndat* = 1000, 500;
- ○ trajectory of random generator *seed* = any integer;
- ○ number of initial model components *ncomi* = 30, 10, 5;
- ○ number of iterations of the initialization algorithm *niter* = 3, 5.

For example, the decision maker sets a value of number of components of the simulator $ncom = 20$. The following results, see Fig. 3, are obtained. Note the difference between Fig. 2 (top left) and Fig. 3 (top left).

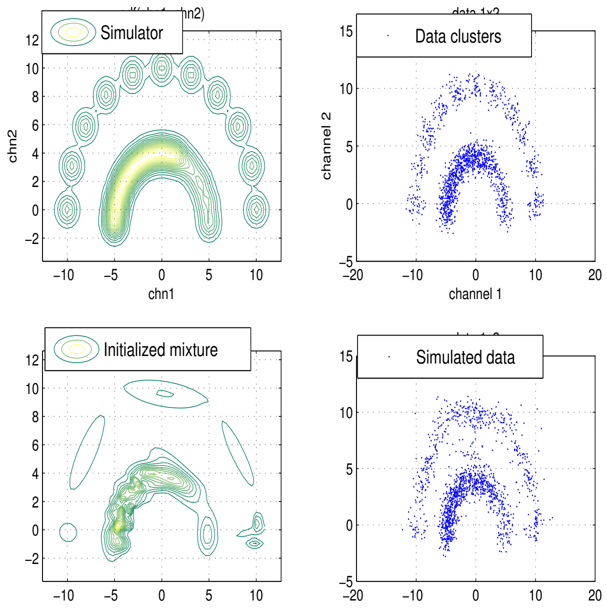


Figure 3: Simulator with $ncom = 20$, data clusters, initialized mixture and data, generated by it

The following dialog appears

```
ENTER:
continue I: initial values S: stop C: >
```

The decision maker chooses among the options:

- ○ *I* – continue the experiment with $ncom = 20$;
- ○ *S* – return the default value $ncom = 40$;
- ○ *C* – stop the experiment.

The given example acquaints the decision maker with the task of mixture initialization and enables him to learn the simple settings of the presented algorithm. It means, that at the given stage the *experience* of the decision maker includes the mixture model structure.

The next decision-making task to be solved is to estimate the mixture parameters, or *the mixture estimation*, see Section 2. Let's show this task on another case study with another mixture, which not only familiarizes the decision maker with the mixture estimation, but also enables to compare different algorithms of the estimation.

4.2 Example 2

The title of the case study at the main page of the system is: *Comparison of Mixture Estimation Algorithms – Static Case*. The aim of the example is to compare results of the approximate estimation of static mixture parameters with the help of *projection algorithm*, *quasi-Bayes*, *batch quasi-Bayes* and *forgetting branching* algorithms.

The PDF format page of the case study contains the following information:

System: ndat two-dimensional data are generated by the static *normal mixture* with *ncom* components. Random generator is determined by *seed*. The components have common diagonal covariance with the diagonal filled by *diac*. Their centers are uniformly positioned on the unit circle. The component weights are linearly decreasing;

Experience: data; mixture structure;

Ignorance: mixture parameters;

Admissible strategy: formulated only for design tasks, here it is not formulated;

Loss function: negative v-*log-likelihood*, normalized by the number of the processed data;

Decision: estimation of mixture parameters.

The static mixture with $ncom = 3$ components, whose centers are uniformly positioned on the unit circle, is taken as a simulator, see Fig. 4 (top left). The simulator generates data, and the mixture estimation is done. Three algorithms of the mixture estimation are used here: the projection-based algorithm, see results at Fig. 4 (top middle); quasi-Bayes estimation algorithm, see Fig. 4 (top right); batch quasi-Bayes algorithm, see Fig. 4 (bottom left), and forgetting branching algorithm, see Fig. 4 (bottom middle). The performance of algorithms is influenced by the number of iterations *niter* and forgetting rate *frg*. The forgetting branching technique uses additionally the alternative small forgetting rate *frga*.

Negative v-*log-likelihood*, normalized by the number of the processed data, is the loss function. Thus, it is the main criterion, except the visual verifying, to judge, which results were the best ones. In the previous example the form of the mixture model was very easily verified, but at the present case it is not so obvious. The values of the v-*log-likelihood* are shown at Fig. 4 (bottom right). It can be seen, that the value of the v-*log-likelihood* for quasi-Bayes algorithm is the highest one.

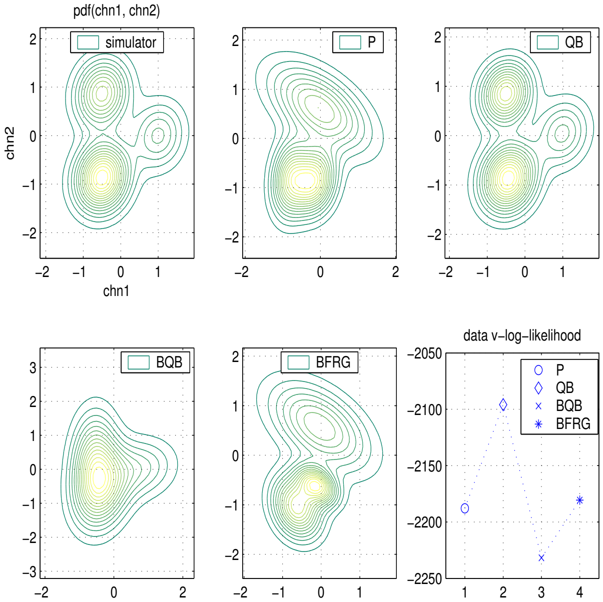


Figure 4: Simulator, estimation with the projection, quasi-Bayes, batch quasi-Bayes and forgetting branching algorithms, v-log-likelihood

The button "function help" gives here the description of Mixtools function *mixest* and its parameters, which include the choice of the desired method of estimation. The button "run example" starts $MATLAB$, which produces the dialog, similar to the previous case study, see Section 4.1:

```
--- System ---------------------------
number of components      ncom  = 3;
diagonal of noise cov.    diac  = 0.2;

--- Data -----------------------------
length of data            ndat  = 1000;
seed                      seed  = 123;

--- Estimation -------------------
number of repetitions   niter = 10;
forgetting rate               frg   = 1;
alternative forget. rate frga  = 0.61;

any changes?
ENTER = no, or type a command: >
```

The default values are shown. According to recommendations, obtained in the PDF format page by the button “recommended experiments”, it is recommended to observe the performance of the algorithms with the following values:

- number of mixture components *ncom* = 10, 20;
- diagonal of noise covariance *diac* = 0.5, 1;
- data size *ndat* = 500, 300, 200;
- trajectory of random generator *seed* = any integer;
- number of iterations *niter* = 10, 30, 100;
- forgetting factors *frg* = 0.99, 0.999; *frga* = 0.5, 0.9.

After making the experiments the decision maker sees the final dialog, similar to the shown in Section 4.1.

The presented examples concern the tasks, where the aim of the decision maker is to *describe* the system. Let’s demonstrate the case study for the decision-making task of *control design*, when the decision-making aim is to *influence* the system.

4.3 Example 3

The title of the case study is: *Academic Design with Static Mixtures*. The aim of the example is to inspect the typical results of academic control design for various values of the *ideal probability density function* (pdf), chosen by the decision maker. The ideal pdf defines the desired data ranges and preferences for the particular quantities.

The PDF format page of the case study contains the following specification:

System: *ndat* two-dimensional data are generated by the static *normal mixture* with $ncom = 3$ components;

Experience: past data; mixture structure; mixture parameters;

Ignorance: future data;

Admissible strategy: causal mapping on *recommended pointers*;

Loss function: Kullback-Leibler divergence from the ideal pdf;

Decision: recommended pointers to the mixture components, which have to be active.

The static normal mixture with $ncom = 3$ components, see Fig. 5 (top left), is used as the simulator. The academic design results in the pointers, which recommend the mixture components to be active. It provides such probabilities of the components, so that the resulting mixture has minimal *Kullback-Leibler divergence* from the ideal distribution. The decision maker’s chooses the ideal system behavior, represented at Fig. 5 (top right). Figs. 5 (bottom) show the data generated by the system. The optimization is performed, and the optimized components weights, determining the control strategy, are calculated. Fig. 6 shows the optimized mixture (left) and data, generated by it (right).

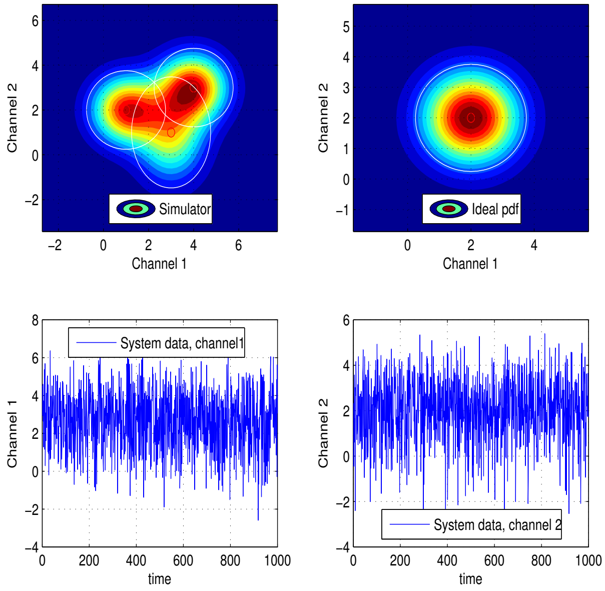


Figure 5: Simulator, decision maker’s ideal, original system data

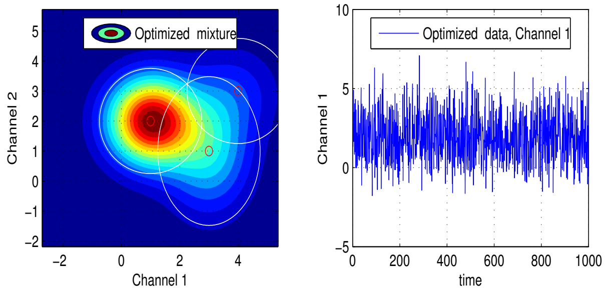


Figure 6: Optimized mixture, optimized data

The button “function help” describes the Mixtools function *aloptim*. $MATLAB$ dialog, starting with the button “run example”, has a structure, similar to the previous examples. The recommendations in the PDF format page include experiments with the optimization with the different decision maker’s ideals. The mean of the ideal pdf can be set to be equal to:

- the mean of the system components *Uth1* = 3; *Uth2*

= 1;

- an approximate center of the system components *Uth1* = 2.5; *Uth2* = 1.5;
- the value outside the limits of the system components means *Uth1* = 0; *Uth2* = 4.

The influence of the changes of covariances for the particular quantities can be tested as well.

After making the experiments the decision maker sees the final dialog, see Section 4.1.

For the sake of simplicity, the examples, concerned only the static mixtures, were chosen for presentation in the paper. The educational system offers the analogous ones, together with a variety of others, for the dynamic tasks.

5 Conclusion

The paper described the interactive educational system for Bayesian decision-making, developed with the help of tools of PDFLATEX and $MATLAB$. The system enables to familiarize with the tasks of Bayesian decision-making, related to the *mixture* models, and to learn the theoretical and algorithmic aspects of their solution.

The system is intended for the postgraduate students and researchers, beginning their activity at the area.

To the advantages of the system one may take, first of all, the combination of theoretical solution and software implementation of the tasks. The uniform way of specification of the decision-making tasks is also important positive feature of the system. The property of the universality is one of the advantages of the educational system. It means, for the first, the principles of the creating of the system can be used for another scientific area. For the second, any software, similar to $MATLAB$, can be used. For example, the educational system for Bayesian statistics, directed at the undergraduate students of the Faculty of Transportation of the Czech Technical University [8], was based on the same principles with Octave as a software for running the tasks.

Among the drawbacks of the presented educational system one may note the *closed source* code of the algorithms of the toolbox Mixtools. The decision maker only learns to use the functions of the toolbox for working with mixture models.

The educational system is available on web (http://as.utia.cz/softwaretools/mixtools).

Acknowledgements: This work was supported by GA ČR grants No. 201/06/P434 and No. 102/03/0049.

References:

[1] 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.

[2] The Matlab Inc., *Matlab documentation*, vol. MA 01760-2098, Natick, 2000.

[3] P. Nedoma, M. Kárný, T.V. Guy, I. Nagy, and J. Böhm, *Mixtools (Program)*, ÚTIA AV ČR, Prague, 2003.

[4] P. Nedoma and P. Ettler, “Data-based adviser to operators of complex processes”, in *Proceedings of the 2002 IEEE International Conference on Control Applications*, M. J. Grimble, Ed., Glasgow, September 2002, pp. 830–831, IEEE.

[5] Ladislav Jirsa, Anthony Quinn, and Ferdinand Varga, “Identification of thyroid gland activity in radiotherapy”, in *ISBA Eighth World Meeting on Bayesian Statistics*, Valencia, Spain, June 2006, accepted abstract.

[6] J. Kratochvílová and I. Nagy, “Traffic control of microregion.”, in *CMP’04: MULTIPLE PARTICIPANT DECISION MAKING, Theory, algorithms, software and applications*, J. Andrýsek, M. Kárný, and J.. Kracík, Eds., Adelaide, May 2004, pp. 161 – 171, Advanced Knowledge International.

[7] P. Gebouský, M. Kárný, H. Křížová, and M. Wald, “Relevance of quantification in combined lymphoscintigraphy diagnostics of upper limb lymphedema”, *Journal of Nuclear Medicine*, 2005, submitted.

[8] E. Suzdaleva, I. Nagy, L. Pavelková, and M. Kárný, *EduKALIBRE EXAMPLES. (Program)*, ÚTIA AV ČR, Praha, 2005.

[9] J. W. Eaton, *GNU Octave Manual. A high-level interactive language for numerical computations*, 3 edition, ISBN: 0-9541617-2-6.

[10] P. Nedoma, M. Kárný, J. Böhm, and T. V. Guy, “Mixtools Interactive User’s Guide”, Tech. Rep. 2143, ÚTIA AV ČR, Praha, 2005.