Alexander LITVINENKO[1], Orken MAMYRBAYEV[2], Natalya LITVINENKO[2], Assem SHAYAKHMETOVA[2], Andrzej KOTYRA[3]

Extreme Computing Research Center, Center for Uncertainty Quantification in Computational Science & Engineering
King Abdullah University of Science and Technology (1),
Institute of information and computational technologies CS MES RK (2), *Lublin University of Technology (3)*

Application of bayesian networks for estimation of individual psychological characteristics

Abstract. An accurate qualitative and comprehensive assessment of human potential is one of the most important challenges in any company or collective. We apply Bayesian networks for developing more accurate overall estimations of psychological characteristics of an individual, based on psychological test results, which identify how much an individual possesses a certain trait. Examples of traits could be a stress resistance, the readiness to take a risk, the ability to concentrate on certain complicated work.

Streszczenie. Jakościowa oraz kompleksowa ocena potencjału ludzkiego jest jednym z najważniejszych wyzwań dla każdej firmy lub grupy. Sieci Bayesowskie stosowane są do opracowywania dokładniejszych ogólnych oszacowań cech psychologicznych jednostki, w oparciu o wyniki testów psychologicznych, które określają, jak dużo dana osoba posiada pewnych cech. (Zastosowanie sieci Bayesowskich do oceny indywidualnych cech psychologicznych)

Keywords: Bayesian network, graphical probability model, psychological test, probabilistic reasoning
Słowa kluczowe: Sieci Bayesowskie, graficzny model prawdopodobieństwa, test psychologiczny, rozumowanie probabilistyczne

Introduction

In this article we discuss applications of Bayesian network methods for solving typical and highly demanding tasks in psychology. We compute overall estimates of the psychological personality traits, based on given answers on offered psychological tests, as well as a comprehensive study of the social status of the individual, their religious beliefs, educational level, intellectual capabilities, the influence of a particular social environment, etc. We believe that the most optimal mathematical model for solving this problem is a graphical probabilistic model with strongly expressed cause-effect relations. Therefore, we chose the Bayesian network as our model. Advantages of the Bayesian network are as follows:

1) The Bayesian network reflects the causal-effect relationship very well.

2) The mathematical apparatus of Bayesian networks is well developed and thus, there are many software implementations of the Bayesian network methods available.

Bayesian framework is very popular in various kinds of applications: parameter identification, Bayesian update, uncertainty quantification, inverse problems [1], and classification.

Bayesian network is a graphical probabilistic model that represents a set of random variables and their conditional dependencies via a directed acyclic graph [2], [3], [4]. For example, a Bayesian network could represent the probabilistic connections between overall economical situations, average salaries and nationalism in society. It can give recommendations to local governments of which steps to undertake to decrease the level of political tensions. Other promising applications are in Human Resource (HR) departments and in marriage agencies. Bayesian networks, by analyzing psychological properties of each individual, and sociological connections between individuals, may help to select a better group for a certain task, prevent possible conflicts and increase performance.

In this work we will apply Bayesian network [4] to find a more accurate overall estimate each investigated psychological personality trait (PPT). Our mathematical model for the construction of overall estimate is the graphical probabilistic model that reflects probabilistic dependencies between the questions used in psychological tests and the overall estimates of the investigated PPT. Due to the presence of cause-effect relationships we will use Bayesian networks as the graphical probabilistic model [5]. We consider also some of the problems which can typically arise during the computing of the overall estimates. For these problems we describe a step-by-step construction of the Bayesian network and we provide the programming code [6]. In the world of psychological tests, there are special software products that help specialists develop new tests and adapt existing products. The main goals of this work are as follows:

1. to develop principles for constructing the overall estimates of PPT based on the usage of the Bayesian network;

2. to demonstrate the potential of graphical probabilistic models for solving problems of this type on several easy examples;

3. to implement these examples in R programming language;

4. to show the capabilities of Bayesian network for a qualitative analysis of the obtained solution.

The structure of the paper is as follows: In Section 2 we introduce the required notions and definitions. Section 3 is devoted to the problem statement. In Section 4 we consider and solve three different examples. We also list the solution in R-code [7]. Finally, in the conclusion we repeat the main achievements and share our experience.

Notions and definitions

In this section we list the necessary definitions that will be used below. These definitions do not always coincide with definitions used in similar works. There are different reasons for this:

- many terms and definitions in psychology are not yet completely formed;

- the meaning of the proposed concepts does not contradict the common notions in other literature;

- our definitions simplify presentation and reading.

Latency is the property of objects or processes to be in a hidden state, without demonstrating itself explicitly.

Psychological tests is a standardized exercise, which results provide information about certain psychological traits of the individual.

Psychological trait is any stable characteristic of a person. This characteristic can be spiritual, moral, social, and is the real reason for the specific behavior of a given person under certain conditions.

A priori estimate is the estimate, obtained before the experiment, on the basis of expert knowledge, some additional information, or taken as a first approximation.

A posteriori estimate is the estimate obtained after the experiment, based on the results of this experiment.

Graph is a set of vertices (nodes) connected by edges. We can also say that graph G is a pair of sets G = (V, E), where V is a subset of any countable set, and E is a subset of V x V. An oriented graph is a graph with oriented edges.

Graphical probabilistic model is a probabilistic model, where the graph shows the probabilistic relationship between random variables. The vertices of the graph are random variables, and the edges reflect the probabilistic relationships between random variables.

Bayesian network is the finite, oriented and acyclic graph representing the graphical probability model.

In the current work the vertices reflect investigated traits and estimates, and the edges reflect dependencies between traits and estimates.

Table 1. Notation

Notation	Meaning
PPT	psychological personality factor(s)
N	is number of psychological factors
F_j	investigated psychological factors, $j = 1, 2, …, N$
E_j	level how a respondent possesses factor F_j, $j = 1, 2, …, N$
Q_{jk}	is the set of questions, $j = 1, 2, …, N, k = 1, 2, …, M_j$. There is a set of questions $\{Q_{jk}\}$ for each investigated factor F_j
E_{jk}	is a grade, which a respondent received for his answer on question Q_{jk}

Problem statement

Let us construct a psychological test. In this test we investigate traits F_j, $j = 1, 2, …, N$. Here F_1 could be the stress resistance of an individual, F_2 the ability to speak the Spanish language, and so on. Suppose that the current respondent possesses all these traits F_j. The level how this respondent possesses trait F_j will be denoted by E_j. For instance, $E_j = 5$ (for the scale 1-very bad, 2-bad, 3-satisfactory, 4-good, 5 excellent) means that the respondent speaks excellent Spanish. Note that F_j are latent values, i.e., for instance, it is hard to say without testing if the respondent possesses stress resistance or not. These E_j estimates depend on many subjective items such as the quality of the tests, the mental state of the respondent, the psychologist's perceptions, etc.

For each investigated trait F_j there is a set of questions $\{Q_{jk}\}$, $j = 1, 2, …, N, k = 1, 2, …, M_j$ in the test. A respondent receives a grade E_{jk} for his answer on question Q_{jk}, where index j means the j-th trait and index k the k-th question.

A professional psychologist can conduct the dependence between the received grades E_{jk} and the presence of F_j by the given respondent. Such type of a priori data can also be obtained from previous experimental data, theoretical knowledge or psychologist's perception. Later on we will start with a Bayesian network that contains a priori probabilities in each node. Then our algorithm will visit each node and recompute each probability value inside (compute posterior probabilities).

The scales of grades for E_j and E_{jk} are discrete and can be very different. Often, scales are chosen on the basis of the subjective wishes of the psychologist. Note that scales for different values may not coincide. The most common scales are:
- a two-point scale {0,1}, {Yes, No}, {presence, absence}, {true, false};
- a three-point scale {-1, 0, 1}, or, for example, {coward, balanced, brave}, {absence, limited presence, presence}, {0. 1, 2}, etc.
- a five-point scale {1, 2, 3, 4, 5} or {0, 1, 2, 3, 4} etc.
- a ten-point scale {1, 2, ..., 10}.
- a hundred-point scale {1, 2, ..., 100}.

The final aim is to assess the presence of PPT by the respondent, based on the a priori estimates of the psychologist and answers of the respondent on the testing questions. It is necessary to take into account the following points:
- there is the probability of a "slip", i. e., when the respondent occasionally gives a wrong answer (pressed a wrong button on the keyboard).
- there is a certain chance of occasionally guessing the correct answer.

Examples of tests

In this section we consider three typical test examples with increasing complexity. All three examples include tables with a priori estimates, given in a table format, as well as a description of these tables. After each description we formulate three possible quantities of interests (we call these quantities of interests - "Tasks"). After that we demonstrate how these Tasks can be solved with the Bayesian network.

The first test example is simple, the posterior is equal to prior. In the second and third test examples we will consider three Tasks. Each example contains the formulation of quantity which should be computed, settings of all required parameters and the corresponding solution in the R-code.

An example of a test with one question. We consider the simplest situation with one question and one trait (PPT). In this case, the simplest graph consists of two vertices and one edge – one vertex for the question and one vertex for the trait and the edge connect these vertices.

Let us compute the grade E_1 how the respondent possesses the trait F_1. For this PPT F_1 there is only one question Q_{11} in this test. We denote the grade, received by the respondent by answering on this question as E_1. A respondent may possess this PPT with varying degrees. Depending on the primary (a priori) grade of the trait F_1, one can a priori assume how respondents will answer question Q_{11}. Corresponding Bayesian network is in Fig. 1.
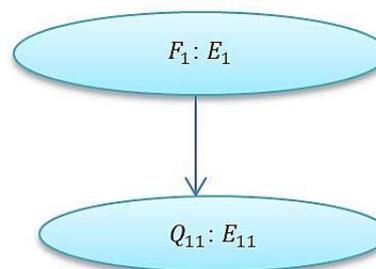

Fig. 1. Bayesian network, where the estimate of the trait F_1 is determined by one question Q_{11}

In this example the overall grade is the same as for the single test question for this trait. There is nothing to compute here. A psychologist usually has some a priori knowledge of how a respondent, who possess a certain trait, can respond to a single test-question for this trait. A priori knowledge is usually obtained from previous

experiments, or relies on the knowledge of a psychologist. Table 2 gives prior estimates for this example. The trait is estimated by a discrete number from the set {1, 2, 3, 4, 5}. The question is estimated also by a discrete value from the set {1, 2, 3, 4, 5}.

One can interpret values from Table 2 in the following way. A psychologist thinks that:

1. 70% of respondents, who do not possess the trait F_1 ($E_1 = 0$) will answer question Q_{11} with grade $E_{11} = 1$.

Table 2. A priori estimates for the test example from Section 4.1, where the overall grade of the trait F_1 is de termined by one question Q_{11}.

Factor assessment E_1 Question assessment E_{11}	0	1	2	3	4
1	70	60	30	10	5
2	20	15	20	15	5
3	5	15	20	25	10
4	3	7	20	30	30
5	2	3	10	20	50

2. only 20% of respondents, who possess the trait F_1 with grade ($E_1 = 3$) will answer question Q_{11} with grade $E_{11} = 5$.

3. only 2% of respondents, who do not possess the trait F_1 ($E_1 = 0$) will answer question Q_{11} with grade $E_{11} = 5$.

This example is simple, the posterior probabilities are equal to prior probabilities. Table 2 gives us all necessary information. The situation becomes more complicated if there are several questions.

a) An example of a test with two questions

We increase the complexity of the example from Section 4.1, namely, we consider two questions in the test. The PPT is estimated on a two-point scale. The question is estimated on a five-point scale. Therefore, the graph consists of 3 vertices and 2 edges [8] (Fig. 2).

Let us build the estimate E_1 of F_1. For this trait there are two questions Q_{11} and Q_{12} in the test. We denote these estimates of questions by E_{11} and E_{12} respectively. Table 3 gives prior estimates for this example.
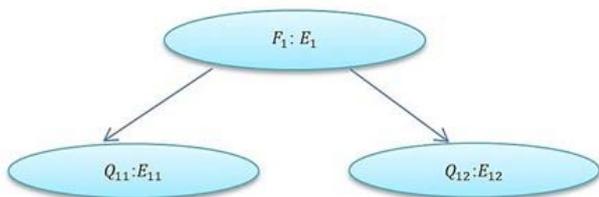

Fig. 2. Bayesian network, where the estimate of the trait F_1 is determined by two questions Q_{11} and Q_{12}

Table 3. Prior estimates for the example from Section 4.2, where the estimate of the trait F_1 is determined by two questions Q_{11} and Q_{12}

Factor assessment E_1 Question assessment E_{11}	Question №1 (Q_{11})		Question №2 (Q_{12})	
	0	1	0	1
1	50	2	60	5
2	30	3	20	5
3	10	5	10	10
4	5	40	5	40
5	5	50	5	40

One can interpret the values from Table 3 in the following way. A psychologist thinks that:

1. 60% of respondents, who do not possess the trait F1 will answer question Q_{12} with grade $E_{12} = 1$.

2. 50% of respondents, who possess the trait F_1 will answer question Q_{11} with grade $E_{11} = 5$. Also 40% of respondents, who possess the trait F_1 will answer question Q_{12} with grade $E_{12} = 5$.

3% of respondents, who do not possess the trait F1 will answer question Q_{11} with grade $E_{11} = 5$. Also 5% of respondents, who do not possess the trait F_1 will answer question Q_{12} with grade $E_{12} = 5$.

Possible quantities of interest in this example could be: What is the probability of receiving the grade E_j for each question by a respondent if a priori probabilities are given as in Table 3? The respondent answered the first question with grade 2, the second question with grade 3. What is the probability that the respondent possesses the trait F_1? The respondent answered the first question with grade 3. What is the probability that the respondent will answer the second question with grades 4 or 5?

To compute these three possible quantities of interest in R environment, we run the following commands as in Algorithm 1. This preprocessing code allow us to include required R packages.

Algorithm 1. R settings

```
#Clear the screen
rm(list=ls(all=TRUE))
#Call the library bioconductor
source("http://bioconductor.org/biocLite.R")
biocLite("RBGL")
biocLite("Rgraphviz")
#Set all libraries we need
install.packages("gRbase")
install.packages("gRain")
library(gRbase)
library(gRain)
library(Rgraphviz)
```

Now we list the required steps in R environment, which set a priori distributions and build preliminary Bayesian network for all three Tasks.

Algorithm 2. A priori parameter settings for the Example from Section "a"

```
#Set a two-point scale for the given factor
lvl ← c("0","1")
#Set a five-point scale for questions
marks ← c("1","2","3","4","5")
#Assume a prior probability that the respondent
#possesses the given factor is 50%
F ← cptable(~F , values=c(50,50), levels=lvl)
#Set a priori probabilities
Q₁₁:F← cptable(~Q11 | F,
  values=c(50, 30, 10, 5, 5, 2, 3, 5, 40, 50),
levels=marks)
Q₁₂:F ← cptable(~ Q12 | F,
  values=c(60, 20, 10, 5, 5, 5, 5, 10, 40, 40),
  levels=marks)
#Plot the graph
cpt.list ← compileCPT(
list(F , Q₁₁.F , Q₁₂.F ))
bnet ← grain(cpt.list)
bnet ← compile(bnet)
plot (bnet$dag)
```
Now we formulate the Task:

Task 1

To compute probability that a random respondent without any a priori knowledge about trait F will answer on 2 questions.
Corresponding R-code, which solves this Task:

```
>    xq1 = querygrain(bnet,
nodes=c("Q11", "Q12"))
>    xq1
$Q11
Q11
1    2    3    4    5
0.26 0.17 0.08 0.23 0.28
$Q12
Q12
1    2    3    4    5
0.33 0.13 0.1  0.23 0.23
```

Result: From this listing in the R environment, one can see that due to prior data (in Table 3), a respondent will answer the first question with grade, for example, 5, with probability 28%, and with grade 3 with probability 8%. Additionally, the last row shows that the respondent will answer the second question with grade 5 with probability 23%, and with grade 3 with probability 10%.
One more task is formulated as follows:

Task 2

Assume that a respondent answered the first question with grade 2, the second question with grade 3. What is the probability that the respondent possesses trait F_1?
Corresponding R-code, which solves this Task:

```
>    bnet.ev <- setEvidence(bnet,
nodes = c("Q11","Q12"),
states = c("2","3"))
xq2 = querygrain(bnet.ev,nodes=c("F"))
>    xq2 = querygrain(bnet.ev,
nodes=c("F"))
>    xq2
$F
F
0    1
0.91 0.09
```

Result: From the last line in the R environment, one can see that the respondent does not possess the trait with probability 91% and possesses the trait with probability 9%.
One more task is formulated as follows:

Task 3

Assume that the respondent answered the first question with grade 2. What is the probability that re-spondent will answer the second question with grade 4 or 5?
Corresponding R-code is:

```
>    bnet.ev <- setEvidence(bnet,
nodes = c("Q11"), states = c("3"))
>    xq2 = querygrain(bnet.ev,
nodes=c("Q12"))
>    xq2
$Q12
Q12
1    2    3    4    5
0.42 0.15 0.1  0.17 0.17
```

Result: The respondent will answer the second question with grade 4 or 5 with probability 17%+17%=34%.

b) An example of test with five questions
In this example we will consider a test with 5 questions. For all 5 questions we set up a five-point scale. The corresponding graph (Fig. 3) consists of 6 vertices (5

vertices for 5 questions and one vertex for PPT) and 5 edges (each edge connects a question with the trait). Let us build the overall estimate E_1 for trait F_1. There are 5 questions $Q_{11}, Q_{12}, \ldots, Q_{15}$ for this trait in the test.
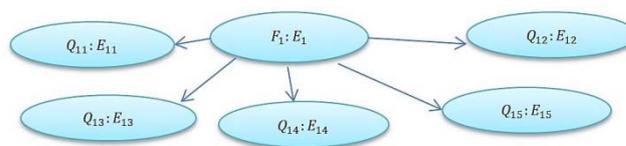

Fig. 3. Bayesian network, where the estimate of the trait F_1 is determined by five questions

We denote estimates of these questions as $E_{11}, E_{12}, \ldots, E_{15}$. Assume that experts, based on personal experience, have compiled Table 4 with prior estimates.
One can interpret the values from Table 4 in the following way. A psychologist thinks that:

Table 4. A priori estimates for the test example from Section 4.3, where the estimate of the trait F_1 is deter-mined by five questions $Q_{11}, Q_{12}, \ldots, Q_{15}$.

Factor assessment E_j Question assessment E_{jk}	0	1	2	3	4
Question №1 (Q_{11})					
1	30	30	10	0	0
2	20	15	20	10	0
3	15	15	10	20	10
4	15	20	30	25	30
5	20	20	30	45	60
Question №2 (Q_{12})					
1	35	25	15	10	0
2	25	20	20	10	10
3	10	25	20	10	10
4	25	15	20	30	30
5	5	15	25	40	50
Question №3 (Q_{13})					
1	40	30	20	15	0
2	20	20	25	10	15
3	20	20	20	15	15
4	20	20	20	40	30
5	0	10	15	20	40
Question №4 (Q_{14})					
1	50	40	35	20	5
2	30	40	30	15	10
3	20	10	25	35	30
4	0	10	10	20	30
5	0	0	0	10	25
Question №5 (Q_{15})					
1	80	50	30	20	10
2	10	20	40	25	15
3	10	20	20	25	35
4	0	10	10	20	20
5	0	0	0	10	20

1. A priori it is known that respondents, who possess trait F_1 with grade $E_1 = 3$ will answer question Q_{11} with grade $E_{11} = 4$ (25%), the question Q_{12} with grade $E_{12} = 4$ (30%), the question Q_{13} with grade $E_{13} = 4$ (40%), the question Q_{14} with grade $E_{14} = 4$ (20%), the question Q_{15} with grade $E_{15} = 4$ (20%).
2. A priori it is known that respondents, who possess trait F_1 with grade $E_1 = 0$ will answer question Q_{11} with grade $E_{11} = 4$ (15%) the question Q_{12} with grade $E_{12} = 4$ (25%), the question Q_{13} with grade $E_{13} = 4$ (20%), the question Q_{14} with grade $E_{14} = 4$ (0%), the question Q_{15} with grade $E_{15} = 4$ (0%).

3. A priori it is known that respondents, who possess trait F_1 with grade $E_1 = 4$ will answer question Q_{11} with grade $E_{11} = 3$ (10%), the question Q_{12} with grade $E_{12} = 3$ (10%), the question Q_{13} with grade $E_{13} = 3$ (15%), the question Q_{14} with grade $E_{14} = 3$ (30%), the question Q_{15} with grade $E_{15} = 3$ (35%).

Possible quantities of interest here could be:
1. What is the probability that a respondent will answer all 5 questions with grade 5?
2. The respondent answered the first question with grade 5, the second and the third questions with grade 4, What is the probability that the respondent has the trait F_1 with grade not less than 3?
3. The respondent answered the first question with grade 5, the second and third questions with grade 3. What is the probability that the respondent will answer the fourth and fifth questions with grades not less than 4?
The program code in R [9], computing quantities of interest, listed above, is the following. The setting commands for R are omitted for brevity.

Algorithm 3. A priory parameter settings for the Example from Section "b"

```
#Set the five-point scale for the given factor
lvl ← c("0","1","2","3","4")
# Set the five-point scale for tests
marks ← c("1","2","3","4","5")
# A priori it is unknown if the respondent possesses
# the factor.
# With probability 20% respondent can possess
# the factor in any amount.
F ← cptable( ~F , values=c(20,20,20,20,20),
levels=lvl)
#Set the a priori data (marks)
Q11.F ← cptable(~Q11 | F, values=c(30,20,15,15,
20,30,15,15,20,20,10,20,10,30,30,0,10,20, 25, 45, 0, 0,
10, 30, 60), levels=marks)
Q12.F ← cptable(~Q12 | F, values=c(35,25,10, 25, 5,
25, 20, 25, 15, 15, 15, 20, 20, 20, 25, 10, 10, 10, 30, 40,
0, 10, 10, 30, 50), levels=marks)
Q13. F ← cptable(~Q13 | F, values=c(40, 20, 20, 20, 0,
30, 20, 20, 20, 10, 20, 25, 20, 20, 15, 15, 10, 15, 40, 20,
0, 15, 15, 30, 40), levels=marks)
Q14. F ← cptable(~ Q14 | F, values=c(50, 30, 20, 0, 0,
40, 40, 10, 10, 0, 35, 30, 25, 10, 0, 20, 15, 35, 20, 10, 5,
10, 30, 30, 25), levels=marks)
Q15. F ← cptable(~Q15 | F, values=c(80, 10, 10, 0, 0,
50, 20, 20, 10, 0, 30, 40, 20, 10, 0, 20, 25, 25,20, 10, 10,
15, 35, 20, 20), levels=marks)
#Plot the graph
cpt.list ← compileCPT(list(F, Q11.F, Q12.F,
Q13.F, Q14.F, Q15.F))
bnet ← grain(cpt.list)
bnet ← compile(bnet)
plot (bnet$dag)
```

Task 4
Compute the probabilities that a respondent with no a priori information will answer all 5 questions.
Corresponding R-code, which solves this Task:
```
> xq1 = querygrain(bnet,
nodes=c("Q11","Q12","Q13","Q14","Q15"))
> xq1
$Q11
```

Q11

1	2	3	4	5
0.14 0.13	0.14	0.24	0.35	

$Q12

Q12

1	2	3	4	5
0.17 0.17	0.15	0.24	0.27	

$Q13

Q13

1	2	3	4	5
0.21 0.18	0.18	0.26	0.17	

$Q14

Q14

1	2	3	4	5
0.30 0.25	0.24	0.14	0.07	

$Q15

Q15

1	2	3	4	5
0.38 0.22	0.22	0.12	0.06	

The output of the R program can be interpreted as follows:
1. A random respondent will answer the first question with grade {1, 2, 3, 4, 5} with probability {0.14, 0.13, 0.14, 0.24, 0.35} respectively.
2. A random respondent will answer the second question with grade {1, 2, 3, 4, 5} with probability {0.17, 0.17, 0.15, 0.24, 0.27} respectively.
3. A random respondent will answer the third question with grade {1, 2, 3, 4, 5} with probability {0.21, 0.18, 0.18, 0.26, 0.17} respectively.
4. A random respondent will answer the fourth question with grade {1, 2, 3, 4, 5} with probability {0.30, 0.25, 0.24, 0.14, 0.07} respectively.
5. A random respondent will answer the fifth question with grade {1, 2, 3, 4, 5} with probability {0.38, 0.22, 0.22, 0.12, 0.06} respectively.

Task 5
The respondent answered the first question with grade 5, the second and third questions with grade 3, the fourth question with grade 2, the fifth question with grade 3. What is the probability that the respondent has the trait F1 with grade not less than 3?
Corresponding R-code, which solves this Task:
```
>bnet.ev <- setEvidence(bnet, nodes
=    c("Q11","Q12","Q13","Q14","Q15"),
states=c("5","3","3","2","3"))
> xq2 = querygrain(bnet.ev,
nodes=c("F"))
> xq2
$F
```
F

0	1	2	3	4
0.05	0.36	0.33	0.11	0.14

The results of the R-code can be interpreted as follows: From the last line in R environment one can see that on the basis of a priori data for values of trait {0, 1, 2, 3, 4} we will have corresponding output probabilities {0.05, 0.36, 0.33, 0.11, 0.14}.

Task 6
Assume that a respondent answered the first question with grade 5, the second and third questions with grade 3. What is the probability that the respondent will answer the fourth and fifth questions with grades not less than 4?
Corresponding R-code, which solves this Task:
```
>bnet.ev <- setEvidence(bnet, nodes
= c("Q11","Q12","Q13"),
states = c("5","3","3"))
>    xq3 = querygrain(bnet.ev,
nodes=c("Q14","Q15"))
>    xq3
```

```
$Q14
Q14
1    2      3      4      5
0.29 0.26  0.24   0.15   0.07
$Q15
Q15
1    2      3      4      5
0.34 0.25  0.23   0.19   0.06
```

The results of the R-code can be interpreted as follows: The respondent will answer the fourth question with grade not less than 4 with probability (15%+7%)=22%. The respondent will answer the fifth question no worse than fourth with probability (19%+6%)=25%.

Conclusion

We considered three different examples of psychological tests. The first test consisted of asking one question, the second of two questions and the third of five questions. After we set up all required statistical parameters and priors, we formulated three possible Tasks and offered their solutions in R environment. The solution includes the construction of a Bayesian network for each Task and computing posterior probabilities. We used the constructed Bayesian networks to develop principles for computing the overall grade of the given trait F (for instance, the stress resistance). This overall grade tells us the level of possession of this trait F by the given respondent. We demonstrated the potential of graphical probabilistic models of three simple examples. Finally, we showed the capabilities of Bayesian networks for qualitative analysis of the resulting solution.

Although we considered relative simple examples with just one trait, the offered technique and software can be used in cases with more traits. An example of a test case with more than one trait will be considered in a soon to be published paper but we also did not observe any restrictions or limitations in that work. The number of questions in each test can also be increased. The offered R-code only solves the described examples. However, this R-code can be modified for larger numbers of traits, questions and tests.

Software

We use R programming language for realization of Bayesian networks due to its popularity among applied scientists/statisticians [10-13]. The analogical work can be done in MATLAB [14] with the known software packages: GeNIe & SMILE, http://genie.sis.pitt.edu; OpenBayes, https://github.com/abyssknight/; BANSY3, http://www.dynamics.unam.edu/; or in one of the commercial products: AgenaRisk Bayesian network tool, Bayesian network application library, Bayesia, Bnet.

Authors: PhD Alexander Litvinenko, Extreme Computing Research Center, Center for Uncertainty Quantification in Computational Science & Engineering King Abdullah University of Science and Technology, Kingdom of Saudi Arabia, e-mail: alexander.litvinenko@kaust.edu.sa; PhD Orken Mamyrbayev, Institute of information and computational technologies CS MES RK, Almaty, Kazakhstan, E-mail: morkenj@mail.ru; M.Sc. Natalya Litvinenko, Institute of information and computational technologies CS MES RK, Almaty, Kazakhstan, e-mail: n.litvinenko@inbox.ru; PhD Assem Shayakhmetova, Institute of information and computational technologies CS MES RK, Almaty, Kazakhstan, E-mail: asemshaykhmetova@mail.ru; PhD., Prof. Andrzej Kotyra, Lublin University of Technology, Institute of Electronics and Information Technology, Nadbystrzycka 38A, 20-618 Lublin, Poland, e-mail: a.kotyra@pollub.pl;

REFERENCES
[1] Matthies H. G., Zander E., Rosic´ B. V., Litvinenko, A. Parameter estimation via conditional expectation: a Bayesian inversion, *Advanced modeling and simulation in engineering sciences*, 3:24, (2016)
[2] Ben-Gal I., Bayesian Networks, *John Wiley & Sons Ltd.*, (2008)
[3] Pourret O., Naim P., Marcot B., Bayesian Networks: A Practical Guide to Applications, *John Wiley and Sons Ltd.*, (2008)
[4] Albert J., Bayesian Computation with R, *Springer* (2009)
[5] Tulupyev A., Nikolenko S., Sirotkin A., Bayesian Networks. A probabilistic logic approach, *Nauka*, (2006)
[6] Litvinenko N., Questionnaires recognition during social research, *News of the National Academy of Sciences of the RK, Physical-mathematical series 3,* b(2016), 77–82
[7] Litvinenko N., Integration of r software environment in c# software environment, *News of the National Academy of Sciences of the RK, Physical-mathematical series 2*, a(2016), 123–127
[8] Murrell P., R Graphics, *Chapman & Hall/CRC*, (2006)
[9] Chambers J., Software for Data Analysis: Programming with R, *Springer*, (2008)
[10] Buhovec A., Moscalev P., Bogatova V., Biruychinskaya T., Statistical data analysis in R, *VGAU*, (2010)
[11] Kabacoff R., R in Action, *DMK Press*, (2014)
[12] Wojcik W, Bieganski T., Kotyra A., et al., Application of algorithms of forecasting in the optical fibre coal dust burner monitoring system, *Proc. SPIE*, 3189 (1997), 100-109
[12] Romanowski A., Grudzień K., Garbaa H., Jackowska-Strumiłło L., Parametric methods for ECT inverse problem solution in solid flow monitoring, *Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska*, 7 (2017), nr. 1, 50-54
[14] Murphy K., The Bayes net toolbox for Matlab, *Computing science and statistics*, (2001), 1024–1034