INTERNATIONAL BLACK SEA UNIVERSITY
FACULTY of BUSINESS MANAGEMENT
BUSINESS ADMINISTRATION PROGRAM

Elaboration of Learning Process Quality Control Method by Using Robust Statistical and Design Methods (on Example of Northern Iraqi Universities)

Ahmet Demir

Extended Abstract of Doctoral Dissertation in Business Administration

Tbilisi, 2015

Scientific Supervisor: Irakli Rodonaia

(full name)

Professor, Doctor, at International Black Sea University

(academic title)

(supervisor's signature)

Experts (full name & academic title):

1. Alexander Milnikov Prof. Dr.

signature

2. Salavat Sayfullin Assoc. Prof

signature

Opponents (full name & academic title):

1. Alexandre Ediberidze Prof. Dr.

signature

2. Medea Tevdoradze Prof. Dr.

signature

Introduction

In many cases the requirement for the learning process quality are given in the form : say, “weak” (failed) students can be thought those ones whose grades are less than 60 and the percentage of them should be 30%; “ordinary” (of acceptable level) students are those ones whose grades are between 61 and 95 grades, the percentage of them should be 65%; in latter range so called “middle” level students are those whose percentage is no more than 50% of total number of students (including failed ones) and 20% of “ordinary” student; say the grade of these “middle” students turns out to be 80 (or any other value), so the grade 80 can be considered as a median of grades distribution; “excellent” students are those ones whose grades are above 95% and the percentage of them is 5%. However, in all known papers such distributions were approximated by either the normal distribution or by some another well-known distributions (beta distribution, gamma distribution, Weibull distribution, etc.). But in case of applying normal distribution the adequacy and precision of results strongly depends on the degree of “skewness” and often may not be acceptable. In case of applying other distributions (beta distribution, gamma distribution, Weibull distribution, etc.) the problem of estimating adequate distribution parameters arises. In many cases analytical expression cannot be obtained in close form. Besides, when requirements for quality changes, the corresponding shapes of PDF and CDF functions also change. As a result, it is necessary to use frequently complicated procedures of distribution parameters estimation.

The similar task is commonly met in the area of product quality control . The quality requirement to the product quality may look as follows. The percentage of deviation from required level of some quality parameter must be no more than ±5% in 95 % of the output of the product; in this case the quality of the product is regarded as “excellent”. To be regarded as “acceptable” the product quality must be as follows: deviation from required level of the quality parameter is ±6%-20% in 3% of the output of the product. The product quality is regarded as “unacceptable” (or defective) if there is the deviation of more than 20% (so the percentage of defective production must be no more than 2%).

It is not clear in advance which type of distribution should be used in this case. The above distributions (reflecting quality requirements) are called hereinafter "pattern" distributions (functions). It is desirable that distribution of grades of actual exams or percentage of deviation from requires quality level of a product would be as close to the pattern distribution as possible.

A pattern distribution presents quality requirement for total learning process (which must take into account results of all relevant tests). That is, grades of many subjects (obtained by a group of students in tests held during one of more courses) must match the pattern distribution in order that the group would be regarded as successful and meeting the requirements of learning quality. Of course, it is possible to compare grades of each actual test with the pattern distribution and then summarize the results. But this approach is associated with a large amount of additional and repeated calculations.

Taking into account all the above-mentioned, a new general method of using a unified non-parametric estimation of relevant grades distributions and further application of its results to the evaluation process of learning quality is developed in the thesis. It is important to point out that the method does not require the execution of rather complicated procedures of estimating distribution parameters (mean, standard deviation, third and fourth moments)). The method can be applied to fit grades of various multiple tests and compare them with pattern distribution by using the same unified techniques and algorithms. The approach provides forming of overall quality criterion for all test scores and method of comparing it with pattern quality requirement.

Purpose of the Study

Generally, the proposed research purpose is to develop technique of determination of quality level in education and production when quality level standards are given in some close form, for example, in the form of distribution functions. In this context, the main objective of research are determined as follows:

- To explore thoroughly the area via studying related works.
- To understand and clarify the specific purposes and ideas of quality management in education, manufacturing and business.
- To develop techniques and methods that assist interested parties to reveal quickly and reliably possible problems and drawbacks in this area

- To propose scientifically sound procedures that provide comprehensive and convenient way for investigation of problems detected
- To propose scientifically sound and effective methods of solving problems detected
- To develop a practically available and convenient procedures to solve the above problems with the highest possible level of automation
- The techniques to be developed must be applicable to the wide areas of implementation: education, manufacturing, economics, etc.

Methodology

To achieve objectives set in the thesis, the following techniques and research methods have been used:

- Advanced non-parametrical methods of fitting complex, non-standard or unknown functional dependencies: Generalized Lambda Distribution , Piecewise Cubic Hermite Interpolant Polinomials, Kolmogorov-Smirnov statistics
- Advanced methods of information theory: Kullback-Leibler divergence metrics
- Methods of Design of Experiments (sequential adaptive sampling)
- Methods of Global Optimization (Genetic Algorithm)
- Methods of meta- modeling based on Neural Networks (Generalized Regression Neural Network)
- Programming in MATLAB

The Main Contributions and the Scientific Novelty

- The requirements for production and learning process quality are different in various manufacturing, business and educational organizations. A new approach to fit these requirements and evaluate the closeness of realistic (actual) quality of production or learning processes (based on quality indicators of output or scores of examination tests) is proposed in the thesis. The technique uses the strictly defined approximation procedures and allows users automatically evaluate of closeness of actual quality level when changing quality requirements.

- In case of significant difference between actual and pattern distributions a new approach (using neural network of 'Generalized Regression Neural Network' type) of determining the relevant values of the factors that will bring the actual distribution to the pattern one is proposed in the thesis. Two new and original procedures have been developed in the thesis:
    - the procedure of finding relevant values of integrated factors (parameters) of the overall quality level of learning process. The procedure is based on the genetic algorithm (GA) and is efficient to determine and analyze quickly the overall (integrated) quality level and critical values of relevant parameters of learning process at an educational institution. In case of unsatisfactory quality level, revealed by the procedure, relevant actions to improve overall quality are proposed
    - the GA based procedure does not discriminate between subjects which may be characterized by different optimal values of the parameters and cannot take into account the specific features (peculiarities) of different subjects. For example, the average number of hours that each student has spent on home assignments may be different for, say, mathematics and history. To cope with this problem another procedure, which can determine unsatisfactory quality level of learning process for separate single subject and propose the relevant actions to improve quality of this given subject, has been developed in the thesis.

Both procedures allows users to work in intensive interactive mode (when the performance requirements of quality management processes are not clearly formulated enough) or in completely automatic mode.

It should be noted that the proposed techniques and procedures are applicable to wide range of areas: quality management of learning processes in education, products' quality level in manufacturing, business processes in economics, etc.

Practical Implications and Importance

The main purpose of the research is to produce a relevant framework that could be adopted for quality management in various areas: education, manufacturing, business. The contributions of the research is intended to be the main stream in research of quality management. The presented work can be further developed for provision of reliable and practically convenient methods in the above-mentioned areas.

Structure and volume of the work

The volume of the thesis is 104 pages and consists of 3 chapters, a list of references and list of figures and list of tables.

Definition of the Problem

Given a random sample $x_1, x_2, x_3, \ldots x_n$, the basic problem in fitting a statistical distribution to this data is that of approximating the distribution from which the sample was obtained. If it is known, because of theoretical considerations, that the distribution is of a certain type (e.g., a gamma distribution with unknown parameters), then through moment matching, or some other means, one can determine a specific distribution that fits the data. This, however, is generally not the case and, in the absence of any knowledge regarding the distribution, it makes sense to appeal to a flexible family of distributions and choose a specific member of that family. By a flexible family we mean one whose members can assume a large variety of shapes: skewness in either direction, tails that are truncated or extend to infinity on either or both sides, bell-shaped distributions as well as inverted bell-shaped ones. A second desirable quality for family of distributions to be suitable for fitting is for the family to be able to represent a wide range of distributional characteristics such as moments (or combination of moments) or percentiles (or combinations of percentiles). A third desirable feature would be for the distributions in the family to have convenient, preferably closed form, expressions for at least one of their PDF, CDF, and quantile function.

To provide fitting the wide variety of distribution shapes and to describe data by using a single functional form the approach used in the paper implements the **Generalized Lambda Distribution**

(GLD).The method specifies four parameter values for each case, instead of giving the basic data (which is what the empirical distribution essentially does) for each case. The one functional form allows us to group cases that are similar, as opposed to being overburdened with a mass of numbers or graphs.

The generalized lambda distribution family with parameters λ_1, λ_2, λ_3, λ_4, GLD (λ_1, λ_2, λ_3, λ_4), is most easily specified in

$$Q(y) = Q(y;\lambda_1,\lambda_2,\lambda_3,\lambda_4) = \lambda_1 + \frac{y^{\lambda_3} - (1-y)^{\lambda_4}}{\lambda_2} \qquad (1)$$

where $0 \le y \le 1$. The parameters λ_1 and λ_2 are, respectively, location and scale parameters, while λ_3 and λ_4 determine the skewness and kurtosis of the GLD (λ_1, λ_2, λ_3, λ_4). Recall that percentile function (PF) of the stochastic variable X is the function Q(y) which, for each y between 0 and 1, tells us the value of x such that $F(x) = y$:$Q(y)$ = (The value of x such that $F(x) = y$), $0 \le y \le 1$

Here $F(x)$ is the cumulative distribution function (CDF) of the variable X:

$$F(x) = P(X \le x),\ -\infty < x < +\infty.$$

The restrictions on λ_1, λ_2, λ_3, λ_4 that yield a valid GLD(λ_1, λ_2, λ_3, λ_4) distribution and the impact of λ_3 and λ_4 on the shape of the GLD(λ_1, λ_2, λ_3, λ_4) PDF (Probability Density Function) will be considered later.

It is relatively easy to find the probability density function from the percentile function of the GLD ([6]. For the GLD (λ_1, λ_2, λ_3, λ_4), the probability density function is:

$$f(x) = \frac{\lambda_2}{\lambda_3 y^{\lambda_3 - 1} + \lambda_4 (1-y)^{\lambda_4 - 1}} \qquad (2)$$

at $x = Q(y)$.

As we have seen above, very often the quality requirement are given in the form of required percentiles (percent of failed, ordinary, middle and excellent students, percent of deviation of some product's quality parameters from their nominal values and so on). The percentile-based approach fits a GLD(λ_1, λ_2, λ_3, λ_4) distribution to a given dataset by specifying four percentile-based sample statistics and equating them to their corresponding GLD (λ_1, λ_2, λ_3, λ_4) statistics. The resulting equations are then solved for λ_1, λ_2, λ_3, λ_4, with the constraint that the resulting GLD be a valid distribution

The method, described above, requires usage of the complex tables of various values of parameters λ_3 and λ_4. To automate the fitting process the algorithm P-KS ([7]) is used in the paper. The strategy is to find the set of parameters (λ_1, λ_2, λ_3, λ_4) that give the lowest value of the Kolmogorov-Smirnov estimator EKS :

$$E_{ks} = \max\left|\hat{F}_n - F(x)\right| \tag{3}$$

where $\hat{F}_n$ is the empirical cumulative distribution function (ECDF).

As it was stated above, the pattern distribution is given in the form of some percent. For the example of the section we have the following data (expressed in the form of Matlab statements):

x= [0, 60, 80, 95,100];

y= [0, 0.30, 0.50, 0.95, 1];

In order to form the pattern distribution (with which the actual tests grades should be compared) we need to fit a curve to the given data. The fitted curve will be used to generate data values in intermediate points (other than the original data points) -interpolation points. To provide the smoothness and maximum accuracy of generated data in interpolation points the technique of the shape-preserving cubic splines is used. The plot of the ECDF for pattern distribution looks like (Fig.1)

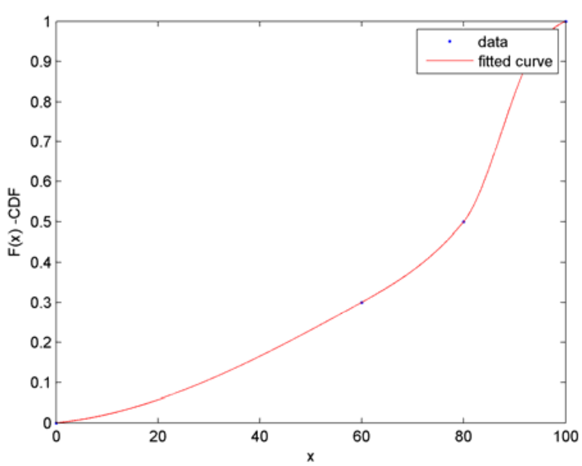


Fig. 1. ECDF for pattern distribution

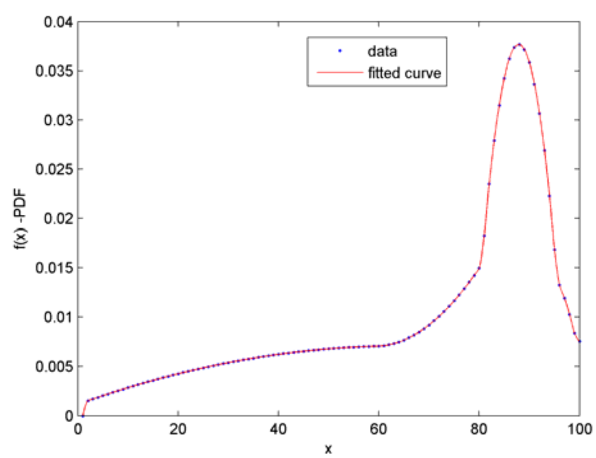


Fig.2 PDF function for pattern distribution

The corresponding PDF function can be obtained similarly and is shown in Fig. 2.

As one can see, the shape of the PDF is non-standard and it is difficult to guess which theoretical distribution can successfully fit it.

Now we can estimate (using relevant Matlab statements) values of the pattern distribution in interpolation points, that is, we can estimate the values of various percentiles (namely, 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th percentiles) of the pattern distribution to be compared with actual Now we can estimate (using relevant Matlab statements) values of the pattern distribution in interpolation points, that is, we can estimate the values of various percentiles (namely, 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th percentiles) of the pattern distribution to be compared with actual tests grades' percentiles. As we stated above, the GLD Percentile-Based Approach to Fitting Distributions intensively uses operations with percentile functions PF (inverse cumulative distribution functions ICDF). We can compute a nonparametric estimate of the inverse CDF. In fact, the inverse CDF estimate is just the CDF estimate with the axes swapped. Here we again use the **Piecewise Cubic Hermite Interpolant Polinomial (PCHIP)** to estimate values of ICDF (Fig.3).

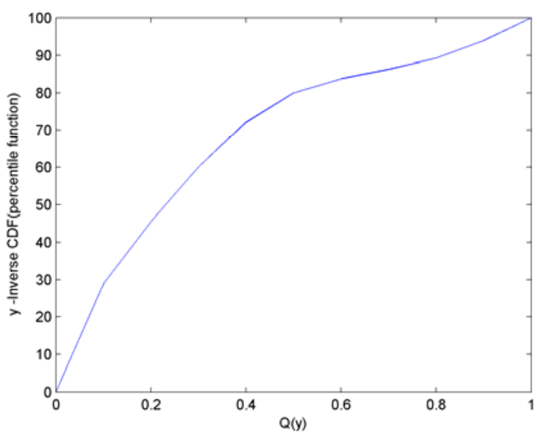


Fig.3 . PCHIP to estimate values of ICDF

Having values of PF we can compute now the values of $\hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3, \hat{\rho}_4$ . . Having computed these values, we now run the procedure P-KS. The solution with the best KS criteria for all possible combinations of pairs (λ_3, λ_4) and associated with them pairs of (λ_1, λ_2) is selected. As it was explained above, knowing λ_1, λ_2, λ_3, λ_4 and using formulas (1) and (2) , we can build the PDF curve: we take a grid of y values (such as .01, .02, .03, . . ., .99, that give us the 1%, 2%, 3%, . . ., 99% points), find x at each of those points from (1), and find f(x) at that x from (2). Then, we plot the pairs (x, f(x)) and link them with a smooth curve.

Now, by using a modification of the **desirability** function, we have to create single integrated PDF curve (which represent PDF curves of all actual tests). For our goals it is enough just to create a single integrated PDF curve by using the arithmetical mean. Suppose that there are PDF curves of R actual tests (given in interpolation points i, namely, i mean points of 10th, 20th, 30th, 40th, 50th, 60th, 70th, 80th, 90th percentiles, see explanation above), denoted $F_r(x_i)$, ($r = 1$ …….R). They are combined to achieve an overall PDF curve D:

$$D(i) = \frac{\sum_{1}^{R} (F_r(x_i)}{R}, \qquad (4)$$

The integrated PDF curve should be compared with the pattern PDF curve obtained above. To determine the closeness (or distinction) of distribution functions (and, thereby, determine the quality of learning process) we'll use Kullback–Leibler Divergence . Let D and P be two PDFs, defined on , where n is $\Re^n$ the dimension of the observed vectors x. The Kullback-Leibler divergence (KL divergence) between D and P is defined as:

$$KL(D \,\|\, P) = \int_{\Re^n} D(x) \log \frac{D(x)}{P(x)} dx \qquad (5)$$

Here D(x) is an integrated PDF, obtained in (4), and P(x) is a pattern PDF.

The problem of obtaining good upper and lower bounds for the relative entropy attracts considerable interest in information theory .We use the following estimation of upper bounds

$$KL(D \,\|\, P) \leq \min\left[\sum_{1}^{n} \frac{D(x_i)^2}{P(x_i)} - 1, \sum_{1}^{n} \sqrt{\frac{D(x_i)}{P(x_i)}} \left|D(x_i) - P(x_i)\right| \right] \qquad (6)$$

If KL metric, computed in (5), is more then value, obtained in (6), we assume that the quality of educational or manufacturing processes does not match the required standards . In this case, relevant actions to improve quality must be undertaken.

Let us assume that comparison of integrated pattern and actual distribution gave us unsatisfactory result: the value (6) is more than the value defined in (5). This means that the quality of learning process is poor and we have to reveal courses and groups that caused this undesired result

Hence, we have to develop a method which can determine courses (or course) whose quality (performance) does not match requirement of the pattern distribution. Besides, we'll examine ways of improving learning quality in these courses.

First of all, we'll consider actual exams. For the simplicity, we consider 5 groups, each containing 20 students (totally 100 students). So, we consider 100 points (grades) obtained in exams for 2 different courses. Moreover, for each exam we consider several factors which can have affect on the quality (that is , on grades obtained). Of course, we assume that such factors are available and can be determined on the basis of interviews of students (filling corresponding questionnaires). Again, for the simplicity we consider the following four factors (in general, number of factors is not crucial for the method developed and any number of factors can be considered):

1. Total midterm evaluation (the vector 'tme') of the student, that is, grades obtained by a student for laboratory works, practical works, quizzes, midterm exam(s) during the current semester; the possible values of this parameter are in the range is: 20÷60; the values of the parameter are filled in the questionnaire by a student.
2. Average number of hours (per week) (the vector 'home_works_hours') that each student has spent on home assignments or home work during the current semester; the possible values are in the range 0.1÷5 hours; the values of the parameter are filled in the questionnaire by a teacher.
3. Average grades (the vector 'aver_prerequizites') that each student has obtained for all prerequisites of the current subject (the vector 'aver_prerequizites'); the possible values are in the range 51÷100; the values of the parameter are filled in the questionnaire by a student.
4. The difficulty level of the exam (the vector 'exam_difficulty'):

1 = No study required

2 = Light revision required

3 = A reasonable effort required

4 = Some real study required

5 = A significant effort requires

The values of the parameter are filled in the questionnaire by a teacher

Let's consider the first factor (parameter). We have the following (sorted) distribution of grades obtained by a student for laboratory works, practical works, quizzes, midterm exam(s):

20 21 23 23 25 25 26 26 26 27 28 29 30 30 31 30 30 31 31 32 32
32 32 33 33 33 30 34 34 35 34 35 36 36 37 37 36 37 37 37 37
38 38 38 37 38 38 38 38 38 39 39 39 40 40 41 41 42 42 41 42 42
43 43 44 44 45 45 45 45 46 45 46 46 47 47 47 48 49 49 49 49 50
51 51 51 52 52 54 53 54 54 55 55 55 56 56 57 59 58

The second factor- the average number of hours (per week) that each student has spent on home assignments or home work during the current semester:

0.49 0.51 0.65 0.65 0.78 0.78 0.86 0.86 0.88 0.9 0.96 1.1 1.1 1.1
1.2 1.2 1.3 1.3 1.3 1.4 1.4 1.4 1.4 1.4 1.5 1.5 1.5 1.5 1.7
1.7 1.7 1.8 1.8 1.8 2 2 2.1 2.2 2.2 2.2 2.2 2.4 2.4 2.5
2.5 2.5 2.5 2.5 2.5 2.6 2.7 2.7 2.7 2.7 2.7 2.8 2.8 2.8 2.8
2.8 2.9 2.9 3.1 3.1 3.1 3.2 3.2 3.2 3.3 3.3 3.3 3.3 3.3 3.3
3.3 3.4 3.4 3.4 3.4 3.5 3.5 3.5 3.6 3.6 3.7 3.7 3.7 3.7 3.7
3.7 3.8 3.8 3.8 3.8 4.1 4.2 4.5 4.8 4.8 4.9

The third factor - average grades that each student has obtained for all prerequisites of the current subject:

51 51 51 52 52 52 53 54 53 56 57 57 57 57 58 58 59 59 59 58 59
59 60 60 61 61 62 63 62 63 63 67 64 65 64 65 66 65 68 67 67
68 68 68 69 68 69 69 70 69 70 70 71 72 72 72 72 74 74 75 75 75
75 74 76 76 76 77 76 77 77 78 76 79 79 80 80 81 80 81 82 83 82
83 85 84 84 85 85 87 86 87 87 88 88 89 90 95 95 97

The fourth factor - difficulty level of the exam; for all students difficulty level is 3 (medium level).

On the basis of these factors (parameters) the independent training set (the 4x100 array 'independent_training_set') has been formed:

Columns 1 through 12:

20	21	23	23	25	25	26	26	26	27	28	29
0.49	0.51	0.65	0.65	0.78	0.78	0.86	0.86	0.88	0.9	0.96	1.1
51	51	51	52	52	52	53	54	53	56	57	57
3	3	3	3	3	3	3	3	3	3	3	3

Columns 13 through 25:

30	30	31	30	30	31	31	32	32	32	32	33
1.1	1.1	1.2	1.2	1.3	1.3	1.3	1.4	1.4	1.4	1.4	1.4
57	57	58	58	59	59	59	58	59	59	60	60
3	3	3	3	3	3	3	3	3	3	3	3

. . . .

Columns 97 through 100:

56	57	59	58
4.5	4.8	4.8	4.9
90	95	95	97
3	3	3	3

Now we form the dependent training set (the vector 'dependent_training_set'), that is grades obtained by students for one of the actual exams:

12	12	17	18	21	22	23	24	24	24	26	27	28	28	28	30	30	31	31	33
33	34	35	36	37	39	40	40	41	41	43	45	46	46	46	46	46	48	50	50
51	53	54	55	55	55	55	56	56	57	57	57	58	62	63	64	64	65	66	66
66	67	67	67	70	71	72	73	73	74	75	77	77	77	78	78	79	80	80	80
81	81	82	82	82	83	83	84	86	87	88	88	89	89	90	90	92	97	99	99

As one can see, the distribution of grades is as follows: about 50% of students have grades less or equal 60, about 30% of students have grades between 61 and 80, about 18% of students gave grades between 81 and 95, and 2% of students have grades between 96 and 100. This distribution of grades, of course, does not match the pattern (required) distribution

Now we have to perform the following task: to find the dependence of the grades on these factors (parameters) (which form the independent training set) and then try to determine the minimum values of the factors that will bring the actual distribution to the pattern one. That is, percents of students received corresponding grades must match the values required by the pattern distribution. For example, percent of students who received grades less or equal 60 must be 40%, percent of students who received grades between 61 and 80 must be 20%, percent of students who received grades between 81 and 95 must be 30%, and percent of students who received grades between 96 and 100 must be 10%. The percentage of actual grades (see above) is quite different.

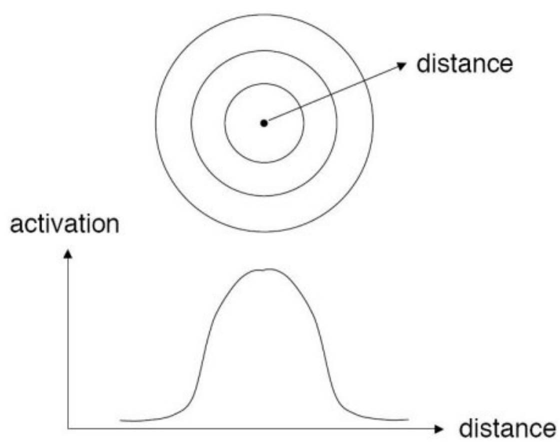
To perform this task there are many difficulties. The character of the dependence of the percent distribution of students received certain marks on these parameters is absolutely unclear. Moreover, the dependencies in our case are likely *non-linear*. Consequently, it is impossible to determine in advance the type of regression dependence, which is necessary to carry out the regression analysis.

Based on the above reasoning, the most adequate approach is the use of the neural networks. Using this approach it is possible theoretically reasonable and objective research and identification of the *hidden* nature of the above dependence. Neural networks - a powerful modeling tool, allowing to reproduce extremely complex dependencies. Neural networks are non-linear in nature. In addition, neural networks can cope with the "curse of dimensionality", which does not allow to simulate non-linear dependencies in the case of a large number of variables. Then, after the determination of this relationship, one can use it to determine the needed values of the parameter. This is the purpose of the proposed approach.

To build neural network model for our task we'll use the *Generalized Regression Neural Network (GRNN)*. It is known the GRNN is a much efficient method for fitting or approximating the complex dependencies. Generalized Regression Neural Networks (GRNN) is a special case of Radial Basis Networks (RBN) [4]. Here a *radial basis function* (RBF) (also called a *kernel* function) is used to predict value of the dependent variable in some point by taking into account the values of dependent variable in neighbor points. The RBF is applied to the distance to compute the weight (influence) for each point. The radial basis function is so named because the radius distance is the argument to the function.

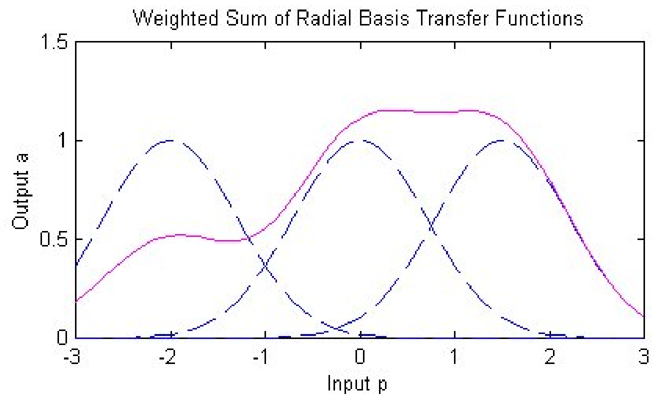
Weight = RBF(*distance*)

The further some other point is from the current point (for which the prediction is being performed), the less influence it has



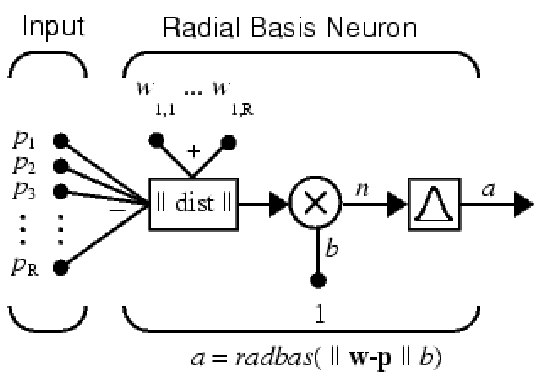
Different types of radial basis functions could be used, but the most common is the Gaussian function. The best predicted value for the current point (for which the prediction is being performed) is found by summing the values of the other points weighted by the RBF function.

The peak of the radial basis function is always centered on the point it is weighting. The sigma value (σ) of the function determines the spread of the RBF function; that is, how quickly the

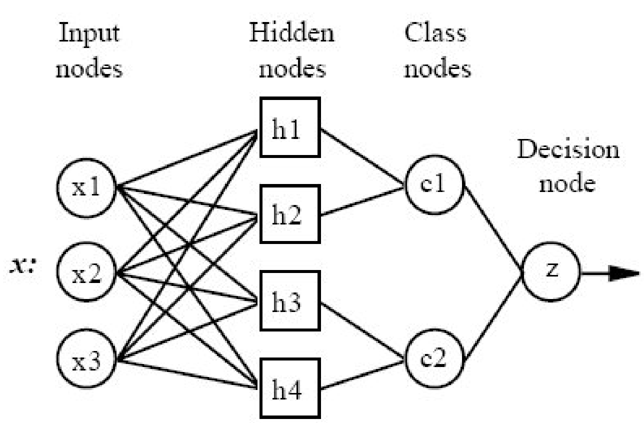


function declines as the distance increased from the point. With larger sigma values and more spread, distant points have a greater influence. If the sigma values are too large, then the model will not be able to closely fit the function. If the sigma values are too small, the model will *overfit* the data because each training point will have too much influence. MATLAB uses the *conjugate gradient* algorithm to compute the optimal sigma values.

Here is a radial basis network with R inputs:



Here is the diagram of GRNN network:



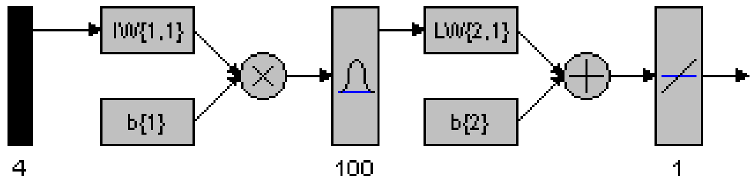
All GRNN networks have four layers:

- **Input layer** — There is one neuron in the input layer for each predictor variable. In the case of categorical variables, *N*-1 neurons are used where *N* is the number of categories. The input neurons (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the interquartile range. The input neurons then feed the values to each of the neurons in the hidden layer.
- **Hidden layer** — This layer has one neuron for each case in the training data set. The neuron stores the values of the predictor variables for the case along with the target value. When presented with the *x* vector of input values from the input layer, a hidden neuron computes the Euclidean distance of the test case from the neuron's center point and then applies the RBF kernel function using the sigma value(s). The resulting value is passed to the neurons in the pattern layer.
- **Pattern layer / Summation layer** —There are only two neurons in the pattern layer. One neuron is the denominator summation unit the other is the numerator summation unit. The denominator summation unit adds up the weight values coming from each of the hidden neurons. The numerator summation unit adds up the weight values multiplied by the actual target value for each hidden neuron
- **Decision layer** — This layer divides the value accumulated in the numerator summation unit by the value in the denominator summation unit and uses the result as the predicted target value.

Unlike standard feedforward networks, GRNN estimation is always able to converge to a global solution and won't be trapped by a local minimum.

We start by calling the command "nntool" of the MATLAB toolbox "Neural Networks". Next we import (using the button 'Import') two datasets: 'independent_training_set' and 'dependent_training_set'. Then we create the neural network of the 'generalized regression neural network' type, the name of the network is GGRN1. Here we assign the spread constant the value 0.7. We use a spread slightly lower than 1, the distance between input values, in order, to get a function that fits individual data points fairly closely. A smaller spread would fit data better but be less smooth.

The network looks like:



As it was mentioned above, the advantage of the GGRN networks is that the training process is carried out in parallel with creation of the network. So, we can immediately use (simulate) the network for the new data.

The GRNN is used in the **procedure of finding relevant values of integrated parameters of the overall quality level of learning process,** developed in the thesis. In this procedure we use the search of the parameters' values that implements the **genetic algorithm (GA)** and the **desirability** functions**.** Namely, the procedure computes the integrated PDF curve. The integrated PDF curve should be compared with the pattern PDF curve obtained above. To determine the closeness (or distinction) of distribution (and, thereby, determine the quality of learning process) we'll use Kullback–Leibler Divergence. Let D and P be two PDFs, defined on $\Re^n$, where n is the dimension of the observed vectors $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

In the thesis we use the following estimate of the "good" upper bound ($UBEst(\mathbf{x})$) for the $KL(\mathbf{x})$:

$$UBEst(\mathbf{x}) = \min\left[\sum_{1}^{n}\frac{D(x_i)^2}{P(x_i)} - 1, \sum_{1}^{n}\sqrt{\frac{D(x_i)}{P(x_i)}}\left|D(x_i) - P(x_i)\right|\right]$$

So, in accordance with the above reasoning, we build for the given vector $\mathbf{x}$ the **fitness function** $FF(\mathbf{x})$:

$$FF(\mathbf{x}) = KL(\mathbf{x}) - UBEst(\mathbf{x})$$

Recall that a vector $\mathbf{x}$ is computed by the implementation of the Generalized Regression Neural Networks (GRNN), which is assumed as a metamodel in the thesis. Let us introduce the following notations:

$\mathbf{x}$ - a solution vector for the GA optimization process

$f(\mathbf{x})$ – the output of the actual fitness function (when output values of actual integrated PDF are used)

$p(\mathbf{w},\ \mathbf{x})$ – the output of integrated fitness function as predicted by the GRNN with weights **w** when the solution $\mathbf{x}$ is used as inputs

$\mathbf{x}^*$ - the best known solution to the GA optimization problem

Parallel to GA optimization problem, there is a training process , which consists of finding the set of weights **w** that minimize an aggregate error measure - mean squared error (MSE).

Suppose that during the search for the optimal values of **x**, the procedure applied to the optimization problem generates a set ALL of solutions **x.** Note that $\mathbf{x}^*$ belongs to the set ALL. Let TRAIN be a random sample of solutions in ALL, such that |TRAIN| ≤ |ALL|. Then we define the training problem as :

$$\text{Min } g(\mathbf{w}) = \frac{1}{|\text{TRAIN}|} \sum_{\mathbf{x} \in \text{TRAIN}} (f(\mathbf{x}) - p(\mathbf{w},\mathbf{x}))^2$$

where **w** is the set of optimization variables of the training problem. Since the training problem cannot be solved until there are at least |TRAIN| solution in ALL, the GA search procedure must initially operate without help of the GRNN by evaluating trial solution **x** using real fitness function (when output values of real (actual) integrated PDF are used). As the GA optimization search advances, the set of ALL solutions becomes large enough so that a suitable training set can be constructed. The training problem is then periodically solved with new TRAIN sets in order to improve the accuracy of the prediction generated by the associated neural network GRNN. The Fig. 4 shows the flowchart of the approach.

Optimizing stage involves a GA to optimize fitness function FF and the corresponding combination values of the independent parameters from the possible solution space. Herein, a possible solution represents a **chromosome.** A chromosome is a string type, which is organized by a sequence of the parameters values for the problem. The individual sites on the chromosome where the parameter values are stored are called **genes**. Genes in the chromosome are formed by the values of the parameters. The chromosome evolve through successive iterations , called **generations**. During each generation , the chromosome are evaluated by a fitness function.

The operational steps are given as follows:

Step 1. Set population size, crossover rate P_C , and mutation rate P_M . Initialize a random population of strings of size l. Choose a maximum allowable generation number t_{max}. Set t=0

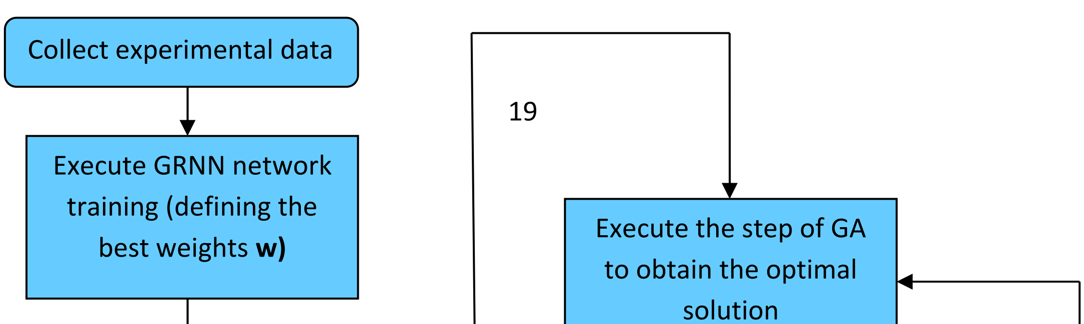


Fig.4 The flowchart of the GA optimization search procedure

Step 2. Calculate the fitness function by inputting parameters values to the trained GRNN

Step 3. If $t < t_{max}$ then terminate

Step 4 Perform reproduction on the population

Step 5. Perform crossover on pair of string with probability P_C

Step 6. Perform mutation on strings with probability P_M

Step 7. Evaluate values of strings. Set t=t+1 and go to the Step 2

Step 8. Obtain the optimal combination values of parameters and the corresponding fitness function FF.

The above procedure uses the Matlab genetic algorithm function **ga** with the syntax

[x fval] = ga(@fitnessfun, nvars, options)

where

- *@fitnessfun* is a handle to the fitness function, where *fitnessfun.m* is an M-file that computes the fitness function
- *nvars* is the number of independent variables for the fitness function.
- options is a structure containing options for the genetic algorithm. If you do not pass in this argument, **ga** uses its default options

The results are given by

- **x** — vector at which the final value is attained
- fval - final value of the fitness function

For our problem the fitness function is FF (the function handle *@FF*, $FF(\mathbf{x}) = KL(\mathbf{x}) - UBEst(\mathbf{x})$), the value of the **nvars** is 4 – amount of the parameters (total midterm evaluation (integrated for all subjects) of the student, average (integrated) number of hours that each student has spent on home assignments, average grades that each student has obtained for all prerequisites of the integrated subject, difficulty level of the integrated exam).

The procedure executes and the final value of the fitness function when the algorithm terminated is *fval* =0.02184 – rather close to the theoretically computed value 0.01851.

The final vector **x** in this example is [16; 1.26; 54; 3]. Recall that the components of the vector **x** are:

x (1) - total midterm evaluation (integrated for all subjects) of the student (=16);

x (2) - average (integrated) number of hours that each student has spent on home assignments (=1.26)

x (3) - average grades that each student has obtained for all prerequisites of the integrated subject (=54)

x (4) - difficulty level of the integrated exam (=3)

As one can see from the details of the GA optimization procedure, this approach is efficient to determine and analyze quickly the overall quality level and critical values of relevant parameters of learning process at an educational institution. In fact, values only of integrated indicators are considered in the GA procedure. In case of unsatisfactory values, obtained by the procedure, relevant actions to improve quality must be undertaken. However, this approach has some drawbacks. The matter is that different subjects may be characterized by different optimal values of the parameters. For example, the average number of hours that each student has spent on home assignments may be different for, say, mathematics and history. The proposed GA procedure does not discriminate between such cases and cannot take into account the specific features (peculiarities) of different subjects. So, the GA procedure cannot propose the actions that may improve the quality of learning process for a single subject. Of course, the procedure has a variant that operates with non-integrated PDF (PDF for a single subject), but it lacks some advantages which has the integrated variant of the procedure.To cope with this problem another procedure, which can determine unsatisfactory quality level of learning process for a single subject and propose the relevant actions to improve quality of this given subject, has been developed in the thesis

In the alternative **sequential adaptive procedure** the quality level of learning process for the single subject is being determined. The proposed procedure essentially uses the principles of sequential adaptive strategy with sequential sampling of experimental points. To fulfill our goal (to determine the minimum values of the factors that will bring the actual distribution of test grades (of **single subject**) to the pattern one) we can try to change the values of one of the factors (or all factors). The change may consist in increasing or decreasing of the factor (depending on whether the percentage of actual grades is more or less than the percentage of the pattern one). For the percentage of failed students (those who obtained less than 60 grades) the changes are as follows: if the actual percentage is more than pattern one, the algorithm has to increase the values of three first factors (total midterm evaluation of the student, average number of hours that each student has spent on home assignments, average grades that each student has obtained for all prerequisites of the current subject) and, maybe, to reduce the difficulty level of the exam (in case if changes of first three factors did not help). The order of factors (priorities) that must be changed is determined by the administration. One option is the priority is as follows:

1. total midterm evaluation of the student
2. average number of hours that each student has spent on home assignments
3. average grades that each student has obtained for all prerequisites of the current subject
4. difficulty level of the exam

Another option is when all factors have the same priorities.

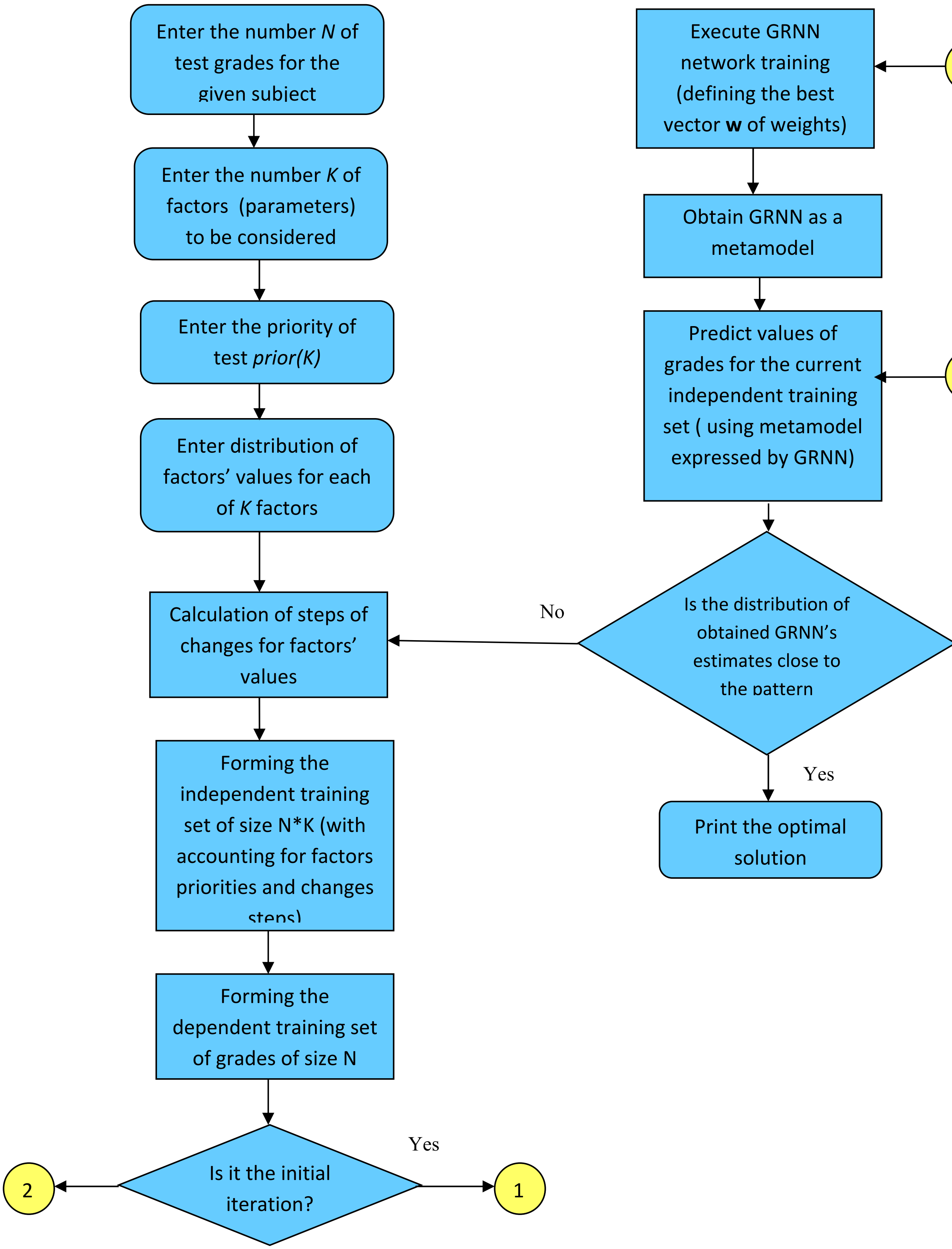
For other percentages (percentages of students have grades between 61 and 80, percentages of students gave grades between 81 and 95, and percentages of students have grades between 96 and 100) the rule is as follows: if the actual percentage is less than pattern one, the algorithm has to increase the values of three first factors maybe, to reduce the difficulty level of the exam (in case if changes of first three factors did not help). If the actual percentage is more than pattern one, then maybe it is necessary to increase the difficulty level (since the recommendation of decrease values of first three factors is not acceptable from a pedagogical point of view).

Here it is necessary to emphasize the following point: changing the values of factors is intended to determine the values which may be useful in future, that is, the updated values of factors can be taken into account and recommended for preparation to future exams. For example, if the average prerequisite grades for failed students, found by the proposed procedure, is, say, 68, then the administration may issue the decree that students who have the average prerequisite less than 68, cannot be admitted to the exam, otherwise the probability of pattern (required) requirements' violations increases and, thereby, the quality of the educational process deteriorates. Besides, it is assumed that ability of students to learn (and which are fixed by grades obtained in the exams) will be unchanged in future. The main goal of the proposed approach is to meet the requirements of the quality of learning process developed by the university's administration.

The Fig. 5 shows the flowchart of the proposed sequential adaptive procedure.

In accordance with the above mentioned we can continue by creation a new training set, for example, for total midterm evaluation factor. The step of change is: (maximum value –minimum value)/10, or (59-20)/10=3.9. The rounded value is 4. The updated values of the factor is:

24 25 27 27 29 29 30 30 30 31 32 33 34 34 35 34 34 35 35 36
36 36 36 37 37 37 34 38 38 39 38 39 40 40 41 41 40 41 41 41
41 42 42 42 41 42 42 42 42 42 43 43 43 44 44 45 45 46 46 45
46 46 47 47 48 48 49 49 49 49 50 49 50 50 51 51 51 52 53 53
53 53 54 55 55 55 56 56 58 57 58 58 59 59 59 60 60 60 60 60



Enter the number *N* of test grades for the given subject
Enter the number *K* of factors (parameters) to be considered
Enter the priority of test *prior(K)*
Enter distribution of factors' values for each of *K* factors
Calculation of steps of changes for factors' values
Forming the independent training set of size N*K (with accounting for factors priorities and changes steps)
Forming the dependent training set of grades of size N
Is it the initial iteration?
Yes
2
1
Execute GRNN network training (defining the best vector **w** of weights)
1
Obtain GRNN as a metamodel
Predict values of grades for the current independent training set (using metamodel expressed by GRNN)
2
Is the distribution of obtained GRNN's estimates close to the pattern
No
Yes
Print the optimal solution

Fig.5 Flowchart of the sequential adaptive procedure.

Now we again call the GRNN model and submit updated training set. The result (updated values of grades) are:

18	21	22	23	24	24	26	28	26	28	32	33	33	33	35	34	36	36	36	36
36	36	40	41	42	42	41	46	46	46	46	54	46	47	46	47	54	48	56	54
54	57	57	57	57	57	60	60	64	60	64	64	64	67	67	67	67	75	75	72
75	75	77	77	77	77	77	78	77	78	80	80	77	80	80	80	80	82	82	82
82	83	84	86	86	86	86	87	89	89	89	90	90	90	90	90	92	99	99	99

As one can see, the percentage of failed is reduced and now is 46% .

The further action depends on the distribution of priorities among the factors. If all factors have the same priorities then the next action is changing the values of the next factors (here this is average number of hours that each student has spent on home assignments). If the current factor has higher priority, then the its values is increased by the step (equal to 4), the updated independent training set again is submitted to the network, the updated grades are again analyzed and so on. Only if the end of the range of factor's possible values is reached and the desired result is not obtained (that is, no reduction of the percentage of failed students to 40% is obtained), we continue with updating values of the next factor.

Let us assume that all factors have the same priority. In this case we proceed with the next factor.

So, we have to change (increase) the values of the second factor - average number of hours that each student has spent on home assignments. The step of change is: (maximum value –minimum value)/10=0.4441.The updated values of the factor is:

0.93	0.96	1.1	1.1	1.2	1.2	1.3	1.3	1.3	1.3	1.4	1.5	1.5	1.6	
1.6	1.6	1.7	1.7	1.8	1.8	1.8	1.9	1.9	1.9	1.9	1.9	2	2	2.1
2.1	2.1	2.2	2.2	2.3	2.4	2.4	2.5	2.6	2.6	2.6	2.6	2.8	2.9	2.9
2.9	2.9	2.9	3	3	3	3.1	3.2	3.2	3.2	3.2	3.2	3.2	3.2	3.2
3.2	3.4	3.4	3.5	3.5	3.5	3.7	3.7	3.7	3.8	3.8	3.8	3.8	3.8	3.8
3.8	3.9	3.9	3.9	3.9	3.9	3.9	3.9	4	4.1	4.1	4.1	4.1	4.1	4.1
4.2	4.2	4.2	4.2	4.3	4.5	4.6	4.9	5.2	5.3	5.3				

We again submit the updated independent training set to the network, simulate it and obtain the result:

18	21	23	23	24	24	26	28	26	28	32	33	33	33	35	34	36	36	36	36
36	36	40	41	42	42	41	46	46	46	46	54	46	47	46	48	54	48	56	54
54	57	57	57	57	57	61	61	64	61	64	64	64	67	67	67	67	75	75	73
76	76	77	77	77	77	77	78	77	78	80	80	77	80	80	80	80	82	82	82
83	83	84	86	86	86	86	87	89	89	89	90	90	90	90	90	92	99	99	99

As one can see, no reduction of failed students' percentage was obtained: this value remains 46%. Hence , we continue with the next factor - average grades that each student has obtained for all prerequisites of the current subject. We change values of this factor by the appropriate step, again submit updated set to the network, simulate the network, obtain the grades. Now we obtained the reduced percentage of failed students: 42%.

As the required (pattern) value 40% is not reached, we return to the first factor, update it, submit to the network, simulate it and obtain the new result: percentage of failed student is 39%. Hence, we obtain the desired result and it corresponds to the following minimum values of the factors:

Minimum value of the total midterm evaluation =28

Mimimum average number of hours that each student has spent on home assignments=0.93 hours

Minimum average grades that each student has obtained for all prerequisites = 60

Difficulty level of the exam = 3

The combination of values of these factors provides required quality of the learning process in the part of percentage of failed students. The values of the factor can be taken into account when preparing future exams.

As for the other percentages, the search of appropriate values is been performed (with some difference that are describe above).

However, as one can see, this process (using the panel of "nntool" manually) is quite tedious. Therefore, the fully automated module has been developed in the thesis submitted. The MATLAB program constructions were used. Some basic statements of the module are given below.

```
independent_training_set=
[20 21 23 23 25 25 26 26 26 27 28 29 30 30 31 30 30 31 31 32
32 32 32 33 33 33 30 34 34 35 34 35 36 36 37 37 36 37 37 37
37 38 38 38 37 38 38 38 38 38 39 39 39 40 40 41 41 42 42 41
42 42 43 43 44 44 45 45 45 45 46 45 46 46 47 47 47 48 49 49
49 49 50 51 51 51 52 52 54 53 54 54 55 55 55 56 56 57 59 58;
0.49 0.51 0.65 0.65 0.78 0.78 0.86 0.86 0.88 0.9 0.96 1.1 1.1 1.1
1.2 1.2 1.3 1.3 1.3 1.4 1.4 1.4 1.4 1.4 1.5 1.5 1.5 1.5 1.7
1.7 1.7 1.8 1.8 1.8 2 2 2.1 2.2 2.2 2.2 2.2 2.4 2.4 2.5
2.5 2.5 2.5 2.5 2.5 2.6 2.7 2.7 2.7 2.7 2.7 2.8 2.8 2.8 2.8
2.8 2.9 2.9 3.1 3.1 3.1 3.2 3.2 3.2 3.3 3.3 3.3 3.3 3.3 3.3
3.3 3.4 3.4 3.4 3.4 3.5 3.5 3.5 3.6 3.6 3.7 3.7 3.7 3.7 3.7
3.7 3.8 3.8 3.8 3.8 4.1 4.2 4.5 4.8 4.8 4.9;
```

```
51 51 51 52 52 52 53 54 53 56 57 57 57 57 58 58 59 59 59 58
59 59 60 60 61 61 62 63 62 63 63 67 64 65 64 65 66 65 68 67
67 68 68 68 69 68 69 69 70 69 70 70 71 72 72 72 72 74 74 75
75 75 75 74 76 76 76 77 76 77 77 78 76 79 79 80 80 81 80 81
82 83 82 83 85 84 84 85 85 87 86 87 87 88 88 89 90 95 95 97;
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3 3 3 3 3];
dependent_training_set =
[12 12 17 18 21 22 23 24 24 24 26 27 28 28 28 30 30 31 31 33
33 34 35 36 37 39 40 40 41 41 43 45 46 46 46 46 46 48 50 50
51 53 54 55 55 55 55 56 56 57 57 57 58 62 63 64 64 65 66 66
66 67 67 67 70 71 72 73 73 74 75 77 77 77 78 78 79 80 80 80
81 81 82 82 82 83 83 84 86 87 88 88 89 89 90 90 92 97 99 99];
spread = 0.7;
grnn1= newgrnn(independent_training_set, dependent_training_set,spread);
grnn1_outputs=sim(grnn1, independent_training_set);
% below the statement increasing the value of the first factor –total midterm evaluation – by 4 is shown
independent_training_set(1,:)= independent_training_set(1,:) +4;
grnn1_outputs=sim(grnn1, independent_training_set);
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

and so on.

The main advantage of the sequential adaptive procedure is that the tuning of parameters' values is being executed with much higher degree of precision. Besides, the time of execution is, as a rule, significantly shorter than in case of GA optimization procedure.

Conclusions

The problem of evaluation of manufacturing, business and learning processes is defined in the thesis. The need to use non-parametrical approximation methods is demonstrated. A new approach to evaluate the closeness of realistic (actual) quality of production or learning processes to the pattern requirements is proposed. This approach might be used in Manufacturing, Business and Educational fields . The essence of the problem of finding appropriate values or relevant parameters is considered. Basic notions and principles of metamodeling, design of experiments , optimization strategies are also considered. The detailed description of neural networks basics is given. The procedure of finding relevant values of integrated parameters of the overall quality level of learning process and the sequential adaptive procedure to determine quality level of learning process for the single subject are developed and described in the thesis

List of author's publications related to the dissertation:

1. Demir, A.; Rodonaia, I.; Milnikova, I., (2014). On One Approach to Evaluation of Quality Level in Manufacturing, Business and Education, *International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), 2014*, vol., no., IEEE pp.60-63.
2. Demir, A., (2014). Elaboration of Electricity for Production-Consumption Relation of Northern-Iraq for the Future Expectations , *International Journal of Academic Research in Economics and Management Sciences*, Vol. 3, No. 5, Sep. 2014, pp. 101-106.
3. Demir, A & Rodonaia, I. (2014). The Generalized Technique for Evaluation of Quality Level of Manufacturing, Business and Educational Processes. *Journal of Business (IBSUJB)*, *3*(1), 33-36.