

OPTIMAL CONSTRUCTION OF THE PATTERN MATRIX FOR PROBABILISTIC NEURAL NETWORKS IN TECHNICAL DIAGNOSTICS BASED ON EXPERT ESTIMATIONS

In the field of technical diagnostics, many tasks are solved by using automated classification. For this, such classifiers like probabilistic neural networks fit best owing to their simplicity. To obtain a probabilistic neural network pattern matrix for technical diagnostics, expert estimations or measurements are commonly involved. The pattern matrix can be deduced straightforwardly by just averaging over those estimations. However, averages are not always the best way to process expert estimations. The goal is to suggest a method of optimally deducing the pattern matrix for technical diagnostics based on expert estimations. The main criterion of the optimality is maximization of the performance, in which the subcriterion of maximization of the operation speed is included. First of all, the maximal width of the pattern matrix is determined. The width does not exceed the number of experts. Then, for every state of an object, the expert estimations are clustered. The clustering can be done by using the k -means method or similar. The centroids of these clusters successively form the pattern matrix. The optimal number of clusters determines the probabilistic neural network optimality by its performance maximization. In general, most results of the error rate percentage of probabilistic neural networks appear to be near-exponentially decreasing as the number of clustered expert estimations is increased. Therefore, if the optimal number of clusters defines a too “wide” pattern matrix whose operation speed is intolerably slow, the performance maximization implies a tradeoff between the error rate percentage minimum and maximally tolerable slowness in the probabilistic neural network operation speed. The optimal number of clusters is found at an asymptotically minimal error rate percentage, or at an acceptable error rate percentage which corresponds to maximally tolerable slowness in operation speed. The optimality is practically referred to the simultaneous acceptability of error rate and operation speed.

Keywords: technical diagnostics, probabilistic neural network, pattern matrix, expert estimations, clustering, performance maximization.

1. Technical diagnostics based on expert estimations

In the field of technical diagnostics, many tasks are solved by using automated classification [1, 2]. For this, such classifiers like probabilistic neural networks (PNNs) fit best owing to their simplicity [3, 4]. Another merit is that PNNs are relatively insensitive to outliers [5]. The PNN is so simple because it is constructed easily and trained fast. Indeed, to solve a classification problem, only a pattern matrix is required. Each column in this matrix corresponds to a class (i. e., to a state of an object which is under technical diagnostics or surveillance). The elements of the column are features of the object.

To obtain a pattern matrix for technical diagnostics, expert estimations or measurements are commonly involved [1, 2, 6, 7]. Unlike other fields of diagnostics (e. g., in medicine, where a pattern matrix is obtained from images or medical tests), expert estimations are not always reliable and may contain severe biases. Similarly, measurements may be biased due to finite accuracy of tools and methodical inaccuracy. This is why every object state is estimated by at least few experts (or measurements are repeated). Subsequently, a final set of expert estimations is grouped and it can be thought of as if each expert proposes its own pattern matrix. The pattern matrix can be then deduced straightforwardly by just averaging over those expert estimations [8, 9]. However, averages are not always the best way to process expert estimations [10]. Moreover, the PNN at its input can have more than a single representative of a state (class), i. e. a few columns in a pattern matrix can correspond to the same state. Then the PNN performance may be improved by better representing the respective states.

2. Problem statement

Due to the abovementioned reasons of the uncertainty of the PNN pattern matrix deduction, the goal is to suggest a method of optimally deducing the pattern matrix for technical diagnostics based on expert estimations. The main criterion of the optimality is maximization of the PNN performance. However, the subcriterion of maximization of the operation speed should be included as well because the pattern matrix cannot be “stretched” without a limit. Indeed, too “wide” pattern matrices will operate slower. For some technical fields (e. g., where diagnostics is fulfilled frequently), the operation speed is crucial, and thus the slowness will be unacceptable.

3. A general conception of optimizing PNNs

Denote a number of object features by F , and a number of states by S . Then the smallest possible pattern for a PNN is an $F \times S$ matrix. Nevertheless, wider matrices can also be pattern. In general, an $F \times (mS)$ matrix

$$\mathbf{P}(m) = [p_{ij}]_{F \times (mS)}, \quad (1)$$

where $m \in \mathbb{N}$, can be a PNN pattern matrix. In matrix (1), each state is represented with m different patterns (columns), where p_{ij} is an assessment of feature i of the object at state s by

$$s = j - S \cdot \psi\left(\frac{j-1}{S}\right) \text{ for } j = \overline{1, mS} \quad (2)$$

and function $\psi(x)$ returning the integer part of number $x \in \mathbb{R}$ [11].

If there are L experts (group measurements), then $m \leq L$. Those m different pattern matrices can be found from clustering the initial L expert matrices. Obviously, as m increases, the respective PNN operation speed may drop. So, it is necessary to determine an ultimate natural number m_{\max} , at which matrix $\mathbf{P}(m_{\max})$ can be used for the pattern (the slowdown in operation speed will be hard but still tolerable), but matrix $\mathbf{P}(m_{\max} + 1)$ cannot be used for the pattern due to intolerable slowdown in operation speed. This can be done by plotting (tabling) a performance time curve versus m . Instead of real pattern matrices (1) for $m = 1, 2, 3, \dots$, it is sufficient to generate random matrices of size $F \times (mS)$ and train PNNs, whereupon the PNNs are tested (on series of vectors of F numbers, whether they are random or not).

Once a maximally possible size of the pattern matrix is determined, the respective PNNs trained on pattern matrices (1) for $m = \overline{1, m_{\max}}$ are tested. Their performance are plotted (tabled) versus m . Then a number $m^* \in \{\overline{1, m_{\max}}\}$ at which performance is maximal is determined.

So, a general conception of optimizing PNNs is realized via three steps as follows:

1. To determine m_{\max} ($m_{\max} \leq L$).
2. To find m clusters from those L versions of pattern matrix, for each $m = \overline{1, m_{\max}}$.
3. To determine m^* ($m^* \leq m_{\max}$).

Nevertheless, it is worth to additionally note that selection of m_{\max} can be kind of fuzzy. Furthermore, if the performance of a PNN trained on pattern matrix $\mathbf{P}(m^*)$ is not satisfactory, number m_{\max} will be probably increased. This is expected to (at least) slightly affect the operation speed, though.

4. Experimental study

To model generation of the pattern matrix, it is convenient to use normal and uniform randomizers. First of all, a pivot for each state is generated. Denote the pivot value of feature i of the object at state s by \tilde{p}_{is} . So, let

$$\tilde{p}_{is} = \left\lfloor \Psi \left(10 \cdot (\xi_{is} + 1.5\zeta_{is}) \right) \right\rfloor, \tag{3}$$

where ξ_{is} is a random real number drawn from the standard normal distribution (with zero mean and unit variance) for feature i and state s , and ζ_{is} is a random real number drawn from the uniform distribution on interval $(0;1)$, $i = \overline{1, F}$ and $s = \overline{1, S}$.

At the second stage, the pivots are noised by similar randomizers. The noise is equivalent to inaccuracies of measurements and biases in expert estimations. The l -th version of estimation of feature i of the object at state s is

$$\bar{p}_{isl} = \left\lfloor \Psi \left(\tilde{p}_{is} (1 + \sigma_p \xi_{isl}) + 1.5\zeta_{isl} \right) \right\rfloor, \tag{4}$$

where ξ_{isl} and ζ_{isl} are random numbers from the respective standard normal and uniform distributions, $l = \overline{1, L}$, and σ_p is a positive factor of the noise strength. Note that values (4) of expert estimations are integer because a scale for expert estimates is commonly integer or has just a few points. Three examples of generation of pattern matrix and experts' matrices are shown in Figure 1.

| | | | | | | | |
|-----------------|---------|----------|---------|---------|----------|----------|-------------------|
| 8 25 16 | 9 26 13 | 8 27 16 | 8 27 13 | 8 26 15 | 10 22 17 | 7 28 15 | $\sigma_p = 0.1$ |
| 7 8 16 | 6 8 18 | 8 9 16 | 6 9 13 | 7 8 16 | 7 8 14 | 6 10 18 | |
| 1 6 20 | 2 5 21 | 1 5 21 | 2 6 21 | 1 7 24 | 1 7 17 | 2 6 16 | |
| 3 2 26 | 3 2 24 | 4 3 28 | 4 2 28 | 3 2 22 | 3 2 28 | 3 2 26 | |
| 25 10 10 | 21 10 7 | 29 10 11 | 26 9 10 | 27 9 11 | 30 10 10 | 29 10 10 | |
| 0 14 9 | 0 15 8 | 1 4 11 | 0 8 11 | 0 17 4 | 1 14 5 | 0 11 12 | $\sigma_p = 0.25$ |
| 28 3 10 | 32 3 12 | 17 4 10 | 19 3 12 | 25 5 9 | 19 2 10 | 24 2 10 | |
| 1 0 3 | 1 0 2 | 1 0 3 | 1 0 3 | 0 1 4 | 2 1 2 | 2 0 3 | |
| 21 7 5 | 21 8 5 | 23 7 5 | 24 5 4 | 31 7 7 | 25 7 6 | 21 7 3 | |
| 15 1 4 | 17 1 5 | 12 2 5 | 14 2 5 | 10 1 5 | 16 1 3 | 18 1 3 | |
| 4 25 5 | 2 19 3 | 6 30 6 | 3 14 3 | 8 53 9 | 7 34 4 | 1 27 4 | $\sigma_p = 0.5$ |
| 13 30 2 | 15 44 4 | 5 21 2 | 4 83 2 | 20 32 2 | 33 5 4 | 20 42 2 | |
| 25 21 0 | 43 0 1 | 21 13 1 | 34 30 0 | 20 6 0 | 14 36 0 | 4 11 0 | |
| 15 1 9 | 19 1 9 | 23 2 11 | 19 2 4 | 20 2 6 | 13 2 3 | 13 2 5 | |
| 26 5 8 | 36 5 5 | 32 9 1 | 26 6 0 | 30 4 19 | 57 2 1 | 8 10 12 | |

Figure 1. The three examples of generating a 5×3 pattern matrix (highlighted bold on the left) and six experts' matrices ($L = 6$) by increasing the noise strength factor

PNN pattern matrix (1) by (2) is determined as follows:

1. For every state s data

$$\left\{ \left\{ \bar{p}_{isl} \right\}_{i=1}^F \right\}_{l=1}^L \tag{5}$$

are grouped into m clusters. The clustering is done by using the k -means method [12, 13]. Consequently, m centroids of these clusters are found for every state s , $s = \overline{1, S}$:

$$\left\{ \left\{ c_{iks} \right\}_{i=1}^F \right\}_{k=1}^m. \quad (6)$$

2. Matrix (1) is successively formed from centroids (6):

$$p_{iz} = c_{iks} \text{ by } z = s + S \cdot (k-1) \text{ for } s = \overline{1, S}. \quad (7)$$

Once pattern matrix (1) is determined, the respective PNN is trained. Then the PNN is tested using objects whose feature i at state s is

$$q_{is} = \left| \tilde{p}_{is} \left(1 + \sigma_P \xi_{is}^{(1)} \right) + 1.5 \xi_s^{(2)} \right|, \quad (8)$$

where $\xi_{is}^{(1)}$ and $\xi_s^{(2)}$ are another random numbers from the standard normal distribution. It is worth to note that $\xi_s^{(2)}$ implies a normally distributed shift in state s of a test object. This shift is the same for all the features. Thus, model (8) of the test object differs from model (4) of the expert estimation, in which every expert has its “own” shift distributed uniformly. Besides, unlike values (4) of expert estimations, values (8) are not narrowed to a scale or set because they model real-world objects whose features are not tied to any scale.

An example of diagnosing objects with seven features by four states, where 80 expert estimations are involved, is presented in Figure 2 (the PNN testing) and Figure 3 (operation speed). Figure 2 shows that the error rate percentage decreases near-exponentially as the number of clustered expert estimations is increased. Meanwhile, Figure 3 indicates that the PNN operation speed almost linearly decreases. The value of 2.095 % is an asymptotically minimal error rate percentage, and it does not change by $m = \overline{77, 80}$. So, $m^* = 77$ if the respective drop of the operation speed is tolerable (from nearly 87 to 94 seconds, which is about 8 %).

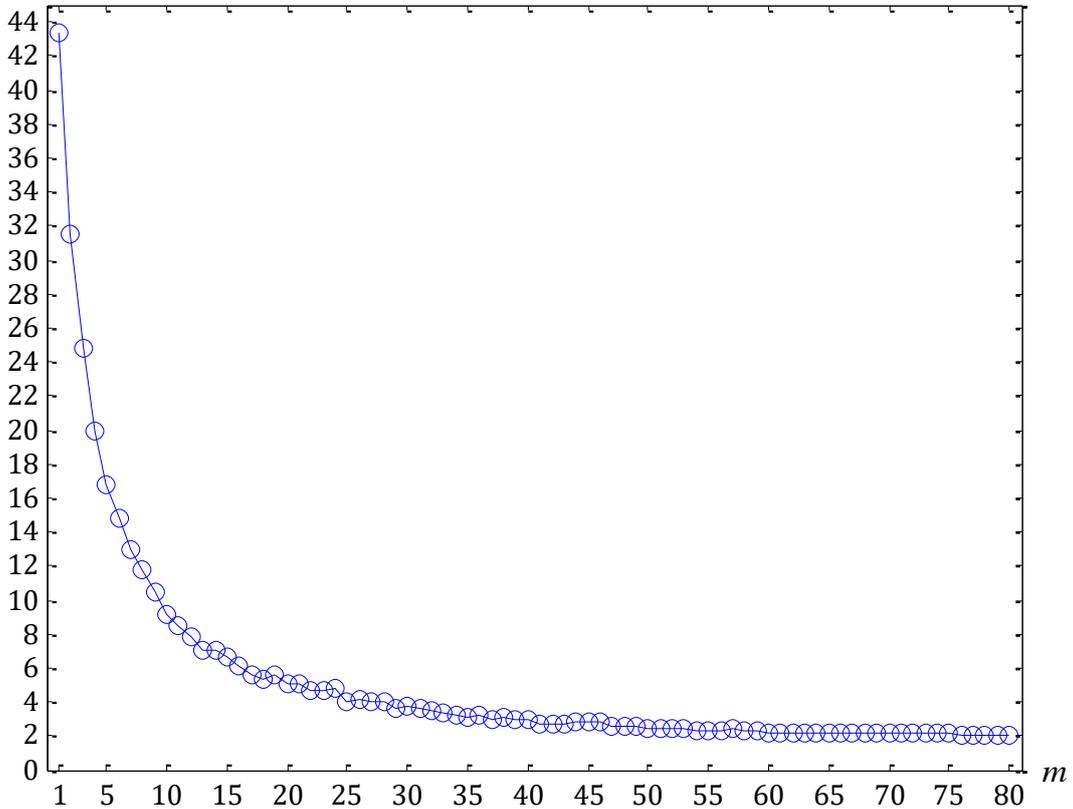


Figure 2. The error rate percentage of PNNs by $F = 7$, $S = 4$, $L = 80$, $\sigma_P = 0.25$, where every object state is tested 10000 times

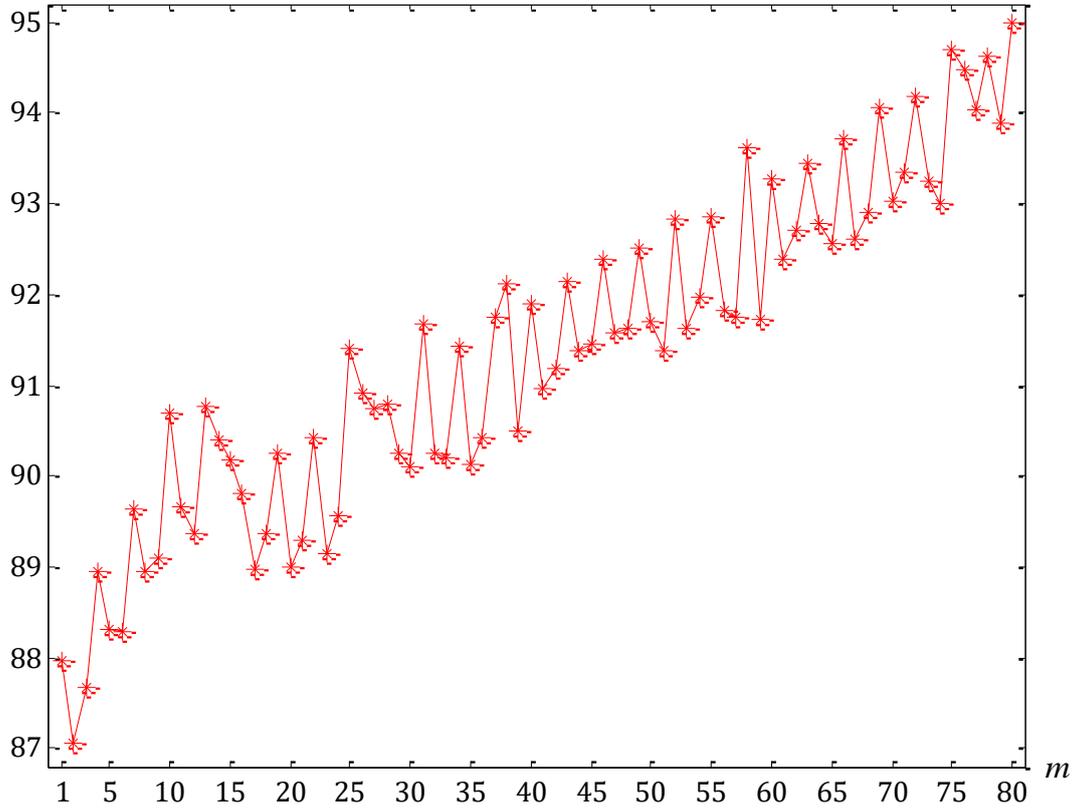


Figure 3. Time (in seconds) spent on testing the PNNs by $F = 7, S = 4, L = 80, \sigma_p = 0.25$ (every object state is tested 10000 times)

Another example, by smaller inaccuracies of measurements and biases in expert estimations, is presented in Figure 4 (the PNN testing) and Figure 5 (operation speed). By the same number of states, having just five features, these PNNs are far faster than those for objects with seven features. The zero error rate percentage (100 % accuracy) is achieved even by forming a pattern matrix as a concatenation of five cluster centroids (for each of the four states). So, in this particular case, $m^* = 5$.

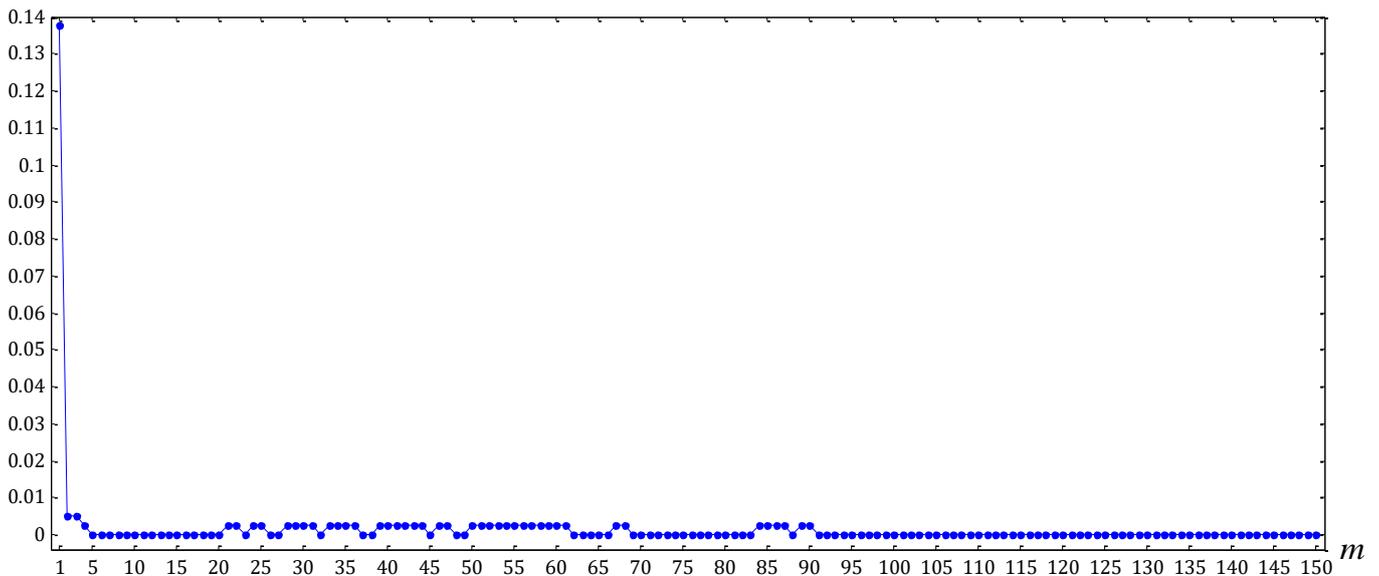


Figure 4. The error rate percentage of PNNs by $F = 5, S = 4, L = 150, \sigma_p = 0.1$ (smaller inaccuracies of measurements and biases in expert estimations than those for Figure 2), where every object state is tested 10000 times

In general, most results of the error rate percentage of PNNs appear to be near-exponentially

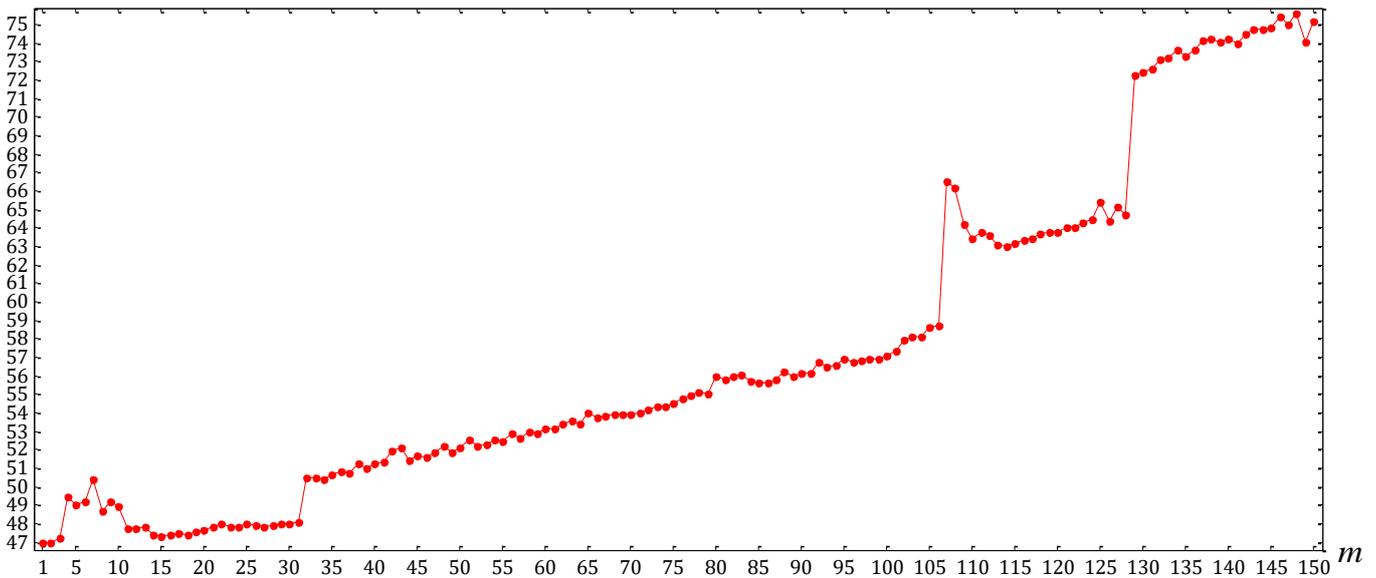


Figure 5. Time (in seconds) spent on testing the PNNs by $F = 5$, $S = 4$, $L = 150$, $\sigma_p = 0.1$ (every object state is tested 10000 times), where computation speed artifacts are observed easier than in Figure 3

decreasing as the number of clustered expert estimations is increased. This also holds true by modeling expert estimations with varying σ_p and other three factors in (4) by (3), and by testing PNNs with object features (8) varying σ_p and the factor at $\xi_s^{(2)}$ as well. Therefore, the example in Figure 2 is a typical performance of a set of PNNs versus the number of clusters per state.

5. Discussion

While the pattern matrix is determined by (5) — (7), which is the general approach, the models of expert estimations and real-world objects are made intentionally specific. In fact, expert estimations are modeled as (4) by (3), and the PNN is tested with object features (8), where only the noise strength factor σ_p is left loose. The specification allows adjusting the models faster owing to the specified factors are close to the best making thus the models highly sensitive (susceptible to small changes in F , S , L , σ_p resulting in drastic changes in the error rate percentage).

The optimal number of clusters is found at an asymptotically minimal error rate percentage, or at an acceptable error rate percentage which corresponds to maximally tolerable slowness in operation speed. However, the pattern matrix cannot be limitlessly “stretched”. The optimality, therefore, is practically referred to the simultaneous acceptability of error rate and operation speed.

6. Conclusion

In technical diagnostics based on expert estimations for using them in PNNs, the pattern matrix is optimally constructed by grouping the estimations for every state into the same number of clusters. The clustering can be done by using the k -means method or similar. The optimal number of clusters determining the PNN optimality is found by the PNN performance maximization. If the optimal number of clusters defines a too “wide” pattern matrix whose operation speed is intolerably slow, the performance maximization implies a tradeoff between the error rate percentage minimum and maximally tolerable slowness in the PNN operation speed.

The suggested optimal construction of the pattern matrix for PNNs can be applied in technical diagnostics of complex objects like devices, buildings, bridges, machines, vessels (watercrafts and

airplanes), etc., based on expert estimations of the object (current) state. Apart from technical and industrial systems, PNNs are nonetheless applicable in other domains (general engineering, social, ecological and economical systems, entertainment, surveillance), where the task is to control the state of objects whose number of features is up to a few tens or hundreds.

References

1. Liu R. et al., “Artificial intelligence for fault diagnosis of rotating machinery: A review”, *Mechanical Systems and Signal Processing*, 2018, vol. 108, pp. 33 — 47.
2. Czichos H., *Handbook of Technical Diagnostics. Fundamentals and Application to Structures and Systems*, Springer, 2013, 566 p.
3. Masters T., “Probabilistic Neural Networks”, in: *Practical Neural Network Recipes in C++*, Academic Press, 1993, pp. 201 — 222.
4. Annema J., *Feed-Forward Neural Networks*, Springer, 1995, 238 p.
5. Romanuke V. V., Yegoshyna G. A., and Voronoy S. M., “Training probabilistic neural networks on the single class pattern matrix and on concatenation of pattern matrices”, *Scientific Papers of O. S. Popov Odessa National Academy of Telecommunications*, 2019, no. 2, pp. 86 — 97.
6. Bose B. K., “Expert systems and applications”, in: *Power Electronics and Motor Drives (Second Edition). Advances and Trends*, Academic Press, 2020, pp. 765 — 788.
7. Tiwari K. and Chong N. Y., “Fusion of information from multiple robots: Fusion of Distributed Gaussian Process Experts (FuDGE)”, in: *Multi-robot Exploration for Environmental Monitoring. The Resource Constrained Perspective*, Academic Press, 2020, pp. 171 — 190.
8. Romanuke V. V., “Fast Kemeny consensus by searching over standard matrices distanced to the averaged expert ranking by minimal difference”, *Research Bulletin of NTUU “Kyiv Polytechnic Institute”*, 2016, no. 1, pp. 58 — 65.
9. Romanuke V. V., “Hard and soft adjusting of a parameter with its known boundaries by the value based on the experts’ estimations limited to the parameter”, *Electrical, Control and Communication Engineering*, 2016, vol. 10, pp. 23 — 28.
10. Bast H. and Weber I., “Don’t compare averages”, in: *Experimental and Efficient Algorithms. Lecture Notes in Computer Science*, vol. 3503, 2005, Springer, Berlin, Heidelberg, pp. 67 — 76.
11. Romanuke V. V., “Heuristic’s job order efficiency in tight-tardy progressive idling-free 1-machine preemptive scheduling of equal-length jobs”, *KPI Science News*, 2020, no. 2, pp. 64 — 73.
12. Hennig C. et al., *Handbook of Cluster Analysis*, Chapman and Hall/CRC, 773 p.
13. Bouveyron C., *Model-Based Clustering and Classification for Data Science (with Applications in R)*, Cambridge University Press, 2019, 446 p.