Soft Comput (2017) 21:199–212
DOI 10.1007/s00500-016-2382-9

METHODOLOGIES AND APPLICATION

Assessment of prediction ability for reduced probabilistic neural network in data classification problems

Maciej Kusy[1] · **Jacek Kluska**[1]

Published online: 5 October 2016

Abstract One of the most important problems in probabilistic neural network (PNN) operation is the minimization of its structure. In this paper, two heuristic approaches of PNN's pattern layer reduction are applied. The first method is based on a *k*-means clustering procedure. In the second approach, the candidates for the network's pattern neurons are selected on the basis of a support vector machines algorithm. Modified models are compared in the classification problems with the traditional PNN, four well-known computational intelligence algorithms (single decision tree, multilayer perceptron, support vector machines, *k*-means algorithm) and PNN trained by the state-of-the-art procedures. Seven medical benchmark databases are investigated and one authors' own real ovarian cancer data set. Comparison is performed on the basis of the global performance indices which depend on the accuracy, sensitivity and specificity. These indices are computed using the standard tenfold cross-validation procedure. On the basis of the reported results, we show that the algorithm based on *k*-means clustering is a better PNN structure reduction procedure. Furthermore, this algorithm is much less time-consuming.

Keywords Probabilistic neural network · *k*-Means clustering · Support vector machines · Medical data classification · Global performance index · Accuracy · Sensitivity · Specificity

Communicated by V. Loia.

✉ Maciej Kusy
mkusy@prz.edu.pl

Jacek Kluska
jacklu@prz.edu.pl

[1] Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, al. Powstancow Warszawy 12, 35-959 Rzeszow, Poland

1 Introduction

Probabilistic neural network, along with multilayer perceptron, radial basis function neural network or self-organizing map is one of the most popular models used in data classification problems. PNN was proposed by Specht (1990) and quickly found many devotees. Its main advantage is that it can quickly learn from input data. Probabilistic neural networks have found their implementation in a variety of classification fields. It was presented in image classification and recognition (Chtioui et al. 1996, 1998; Ramakrishnan and Selvan 2007; Wen et al. 2008), earthquake magnitude prediction (Adeli and Panakkat 2009), multiple partial discharge sources classification (Venkatesh and Gopal 2011), interval information processing (Kowalski and Kulczycki 2014) or medical diagnosis and prediction (Shan et al. 2002; Folland et al. 2004; Huang and Liao 2004; Temurtas et al. 2009; Mantzaris et al. 2011).

From its architecture point of view, PNN is a feed-forward model composed of four layers: an input layer where each element corresponds to a data feature, a radial basis pattern layer which consists of as many neurons as training vectors, a summation layer having single neuron for every class and an output layer that provides the prediction for unknown sample. In the original form, PNN has no weights to be updated; therefore, the training process for this network seems to be feasible. The attention only has to be paid to the appropriate selection and computation of the smoothing parameter for the radial basis neurons. However, the major drawback of PNN lies in the requirement of having one neuron in the pattern layer for each training example (Specht 1992). Thus,

for large data set classification problems, the structure of this model is complex.

In this article, we concentrate on the architecture reduction in the PNN. For this purpose, two alternative approaches of the structure minimization are applied. The first method is based on the application of a *k*-means clustering algorithm to input data in order to determine the optimal number of centroids as the representation of the pattern layer neurons. The second solution consists in the use of a support vector machine procedure which, out of the entire training set, provides the subset of optimal vectors (support vectors) which in turn form the layer of pattern nodes of PNN. Both techniques are tested on the medical data sets. The presented study is a generalization of the results published by the authors in Kusy and Kluska (2013). In that work, the attention is only paid to computing the prediction error on 20 % cases extracted randomly from each of the investigated data sets. Such a solution we consider insufficient. Therefore in this article, the diagnostic accuracy parameters for optimized PNNs are determined by means of a tenfold cross-validation method. The obtained results are compared to the outcomes achieved by the reference classification algorithms: single decision tree, multilayer perceptron, support vector machines and *k*-means clustering procedure and, additionally, to the performance of the state-of-the-art PNN training solutions. Furthermore, we propose the global performance index which has the form of weighted sum of the accuracy, sensitivity and specificity for both binary and multi-class classification problems. The use of such a measure is of a particular importance, especially in medical data classification tasks, as the ones used in this study.

This paper is composed of the following sections. In Sect. 2, we conduct an overview on various PNN reduction methods presented up to this date. Section 3 discusses probabilistic neural network highlighting its basics, structure and a principle of operations. In Sect. 4, the reduction in PNN structure by means of *k*-means clustering and support vector machines algorithms is outlined. In this section, the global performance index is also proposed. Section 5 briefly describes the input data used in this research. The performance of the standard and the modified PNN models is verified in Sect. 6. In this section, we also compare prediction abilities of reduced PNNs with the results obtained by the reference classifiers and the state-of-the-art PNN models. Finally, in Sect. 7, the conclusions are presented.

2 Related work

In general, there exist two categories of studies related to the reduction in PNN construction. The first category includes the clustering techniques. For example, the work reported in Burrascano (1991) presents the learning vector quantization approach for finding representative patterns to be used as neurons in PNN. This method defines a number of examples that are reference vectors which approximate the probability density functions of the pattern classes. The reference in Chtioui et al. (1996) presents the reduction in the size of the training data for PNN by hierarchical clustering. The idea consists in applying the reciprocal neighbors technique, which allows the gathering of examples which are closest to each other. In Zaknich (1997), the quantization method for PNN structure is proposed. The input space is divided into a fixed-size hypergrid, and within each hypercube representative cluster centers are computed. In this way, the number of training vectors in each hypercube is reduced to one. The work presented in Chang et al. (2008) introduces an expectation–maximization method as the training algorithm for PNN. This amounts to the predefinition of the number of clusters as the input data set. A global *k*-means algorithm is used as the solution. In the contribution (Chandra and Babu 2011), an improved architecture for PNN is proposed. The network is designed with an aggregation function based on the *f*-means of training patterns. Such an architecture reduces the number of layers and therefore computational complexity.

In the second category of the studies which focus on the architecture optimization of PNN, the authors utilize non-clustering methods. For example, the model described in Traven (1991) is designed so that it can use far fewer nodes than the training patterns. It is achieved by estimating probability density functions as a mixture of Gaussian densities with varying covariance matrices. In the reference Streit and Luginbuhl (1994), a maximum likelihood algorithm for training the network is presented as the generalization of Fisher method for nonlinear discrimination. It is shown that the proposed PNN requires significantly fewer nodes and interconnection weights than the original model. In Mao et al. (2000), a supervised PNN structure determination algorithm is introduced. This algorithm consists of two parts and runs in an iterative way: smoothing parameter computation by means of genetic algorithm and pattern layer neuron selection. The important nodes for the layer are chosen by employing an orthogonal algorithm. The research presented in Berthold and Diamond (1998) introduces the automatic construction of PNN by the use of a dynamic decay adjustment algorithm. The model is dynamically built during training, which automatically optimizes the number of hidden neurons.

It is important to emphasize that there also exists a third category of articles which are related to the probabilistic neural network. This category encompasses the papers which explore the problem of a smoothing parameter selection as the variable of probability density functions determined for the hidden neurons of the model. Four approaches are usually regarded: single parameter for whole PNN, single parameter for each class, separate parameter for each variable and separate parameter for each variable and class. In the research,

diverse procedures have been developed to solve these tasks (Chtioui et al. 1998; Specht 1992; Mao et al. 2000; Georgiou et al. 2008; Gorunescu et al. 2005; Specht and Romsdahl 1994; Zhong et al. 2007; Kusy and Zajdel 2015).

3 Probabilistic neural network

In this section, the fundamentals of PNN model are presented. Since the principle of operation of this network stems from Bayesian theory, we start with a short description of a Bayes' theorem. Then, it is highlighted how PNN forwards the input signal to succeeding layers to compute its output. The architecture of the network is also shown.

3.1 Bayesian classifier

Assume we are given an observation $\mathbf{x} \in \mathbb{R}^n$ and the number of predefined classes (groups) $g = 1, 2, \ldots, G$. Assume, furthermore, that the probability of the vector $\mathbf{x}$ belonging to the class g equals p_g, the cost (loss) associated with classifying the vector into class g is e_g, and that the probability density functions: $y_1(\mathbf{x}), y_2(\mathbf{x}), \ldots, y_G(\mathbf{x})$, for all classes are known. Then, according to the Bayes theorem, the vector $\mathbf{x}$ is classified to the class g, if

$$p_g e_g y_g(\mathbf{x}) > p_h e_h y_h(\mathbf{x}), \tag{1}$$

for all classes g not equal to h. The probability density function $y_g(\mathbf{x})$ defines the concentration of the data of class g around the vector $\mathbf{x}$. If we also accept that $p_g = p_h$ and $e_g = e_h$, then one can infer that if the probability density function $y_g(\mathbf{x})$ in the neighborhood of $\mathbf{x}$ takes a higher value than $y_h(\mathbf{x})$, the vector $\mathbf{x}$ belongs to the class g.

Unfortunately, in real data classification problems, data set distribution is usually unknown. However, some knowledge on this distribution should be acquired. Therefore, there is a need to determine an approximation of the probability density function $y_g(\mathbf{x})$ computed on the basis of given data. In order to find such an approximation, one often uses the Parzen method (Parzen 1962). The probability density function for multiple variables can then be expressed as follows

$$y(\mathbf{x}) = \frac{1}{l\sigma_1 \ldots \sigma_n} \sum_{i=1}^{l} F\left(\frac{x_{i1} - x_1}{\sigma_1}, \ldots, \frac{x_{in} - x_n}{\sigma_n}\right), \tag{2}$$

where $\sigma_1, \ldots, \sigma_n$ denote standard deviations computed relative to the mean of n variables $x_1, \ldots, x_n$, $F(\cdot)$ is the weighting function which has to be appropriately selected (Masters 1993) and l is the number of input patterns. The Gaussian function is a common choice for weighting in (2) since it is simply well behaved, easily computed and satisfies the conditions required by Parzen's method (Masters 1995).

3.2 Network's structure

From the formula provided in (2), the structure and the operation of PNN are straightforward. It is enough to consider a Gaussian function as the activation for the probability density function and take into account the fact that this function is computed for the examples of class g. This transforms Parzen's definition to the following form

$$y_g(\mathbf{x}) = \frac{1}{l_g (2\pi)^{n/2} \prod_{j=1}^{n} \sigma_j} \sum_{i=1}^{l_g} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{ij}^{(g)} - x_j\right)^2}{2\sigma_j^2}\right), \tag{3}$$

where l_g is the number of examples of class g, σ_j denotes the smoothing parameter associated with jth coordinate, $x_{ij}^{(g)}$ is the jth element of the ith training vector ($i = 1, \ldots, l_g$) which belongs to the class g, and x_j is the jth coordinate of the vector $\mathbf{x}$. The formula presented in (3) provides one of $g = 1, \ldots, G$ summation neurons of PNN structure. The nodes in the preceding layer, called pattern neurons, feed the component to the sum which is measured over each of the examples of the gth class. Hence, l_g hidden neurons constitute the input for the gth summation neuron. Finally, the output layer determines the class for the vector $\mathbf{x}$ in accordance with the Bayes' decision rule based on the outputs of all the summation layer neurons

$$G^*(\mathbf{x}) = \arg\max_g \left\{y_g(\mathbf{x})\right\}, \tag{4}$$

where $G^*(\mathbf{x})$ denotes the predicted class of the pattern $\mathbf{x}$. It can be observed that the pattern layer requires $l = l_1 + \cdots + l_G$ nodes. The architecture of PNN is illustrated in Fig. 1.

4 Proposed algorithms

This section introduces two approaches applied for PNN structure simplification. Both solutions consist in reducing the number of pattern neurons of the network. The first method is based upon a k-means procedure. The second idea, originally hinted in Kluska (2009), truncates the data size by utilizing the support vectors for PNN training. Moreover, in this section the global performance indices are proposed as the indicators for the reduction of the neurons in the network's pattern layer.

4.1 Smoothing parameter selection

In our study, we utilize PNN with single σ for each variable (attribute) and, additionally, for each class. The choice of this

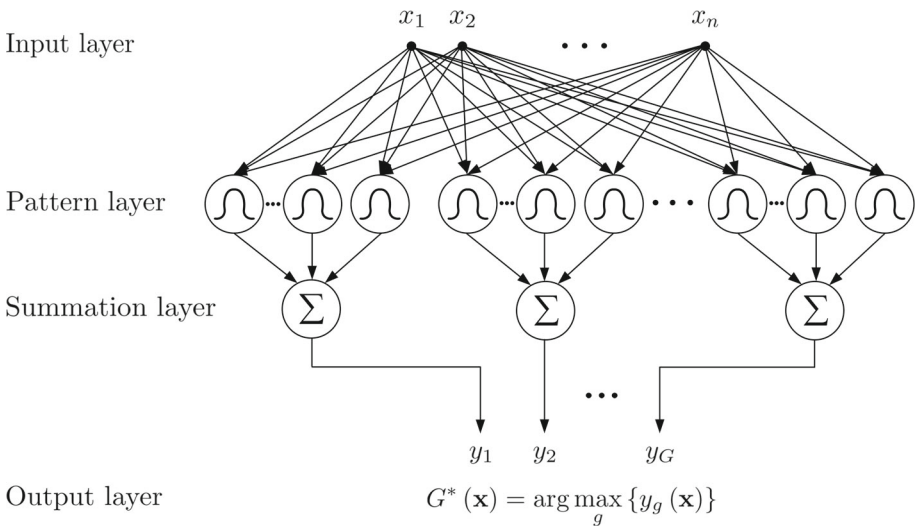

Fig. 1 The architecture of the probabilistic neural network

variant of smoothing parameter selection is the most general approach but imposes, in accordance with formula (3), the inevitability of storing a $G \times n$ matrix of σ's. Henceforth, the gth summation neuron yields to the decision layer the following output signal

$$y_g (\mathbf{x}) = \frac{1}{l_g (2\pi)^{n/2} \prod_{j=1}^{n} \sigma_j^{(g)}} \sum_{i=1}^{l_g} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{ij}^{(g)} - x_j\right)^2}{2\left(\sigma_j^{(g)}\right)^2} \right), \tag{5}$$

where $\sigma_j^{(g)}$ is the smoothing parameter determined for the jth coordinate and gth class. Such an approach gives the possibility of emphasizing the similarity of the vectors belonging to the same class and, simultaneously, improving generalization accuracy.

4.2 PNN structure reduction by the use of k-means clustering

The k-means algorithm is a well-known clustering method developed independently by several researches (Hartigan and Wong 1979; Lloyd 1982), which is considered to be one of the top ten algorithms in data mining (Wu et al. 2008). The basic idea of this algorithm is to try to discover k clusters, such that the records within each cluster are similar to each other and distinct from records in other clusters. The grouping process relies on the iterative minimization of the sum of squared distances computed between input data and the cluster center. An initial set of clusters is defined, and the cluster centers are repeatedly updated until no modification of their coordinate values is obtained. It is worth noticing that there is the difficulty of "optimal" clustering that stems from a huge number of ways to partition a set of l_g patterns into k non-empty clusters. Such a number is a Stirling number of the second kind (Riordan 1958)

$$S\left(l_g, k\right) = \frac{1}{k!} \sum_{i=1}^{k} (-1)^{k-i} \binom{k}{i} i^{l_g}. \tag{6}$$

For example, $S\,(30, 10) \cong 1.7 \times 10^{23}$. Therefore, the first approach in PNN structure reduction utilizes the k-means algorithm in an iterative way. The number of clusters of class g in sth iteration is computed according to the formula

$$i_{s,g} = \text{round}\left(\frac{s}{N} l_g\right), \quad s = 1, \ldots, N-1, \tag{7}$$

where the function round (x) rounds the real positive number x to the nearest integer. In this paper, we assume $N = 10$. It is important to notice that only $i_{s,g}$ pattern layer neurons of class g are involved in the computation of the signal for the summation layer neuron defined in (5).

Let us define the reduction ratio R as a quotient of the number of pattern neurons by the cardinality of the training data set for the PNN

$$R\,(s) = \frac{1}{l} \sum_{g=1}^{G} i_{s,g} \cong \frac{s}{N}, \quad s = 1, \ldots, N-1. \tag{8}$$

Then, the optimal ratio in the sense of the stated problem is $R\,(s^*)$, where in general

$$s^* = \arg\max_s J\,(s)\,, \tag{9}$$

where $J(s)$ is a global performance index defined as follows

$$J(s) = \begin{cases} \alpha \text{Acc} + \beta \text{Sen} + \gamma \text{Spe} & \text{if } G = 2 \\ \sum_{g=1}^{G} \left(\alpha_g \text{Acc}_g + \beta_g \text{Sen}_g + \gamma_g \text{Spe}_g\right) & \text{if } G > 2, \end{cases} \tag{10}$$

where Acc, Sen and Spe denote the accuracy, sensitivity and specificity, respectively, defined as follows

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{11}$$

$$\mathrm{Sen} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \tag{12}$$

$$\mathrm{Spe} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}}, \tag{13}$$

where TP, TN, FP and FN stand for the true-positive, true-negative, false-positive and false-negative counts, respectively (Altman and Bland 1994). Acc, Sen and Spe are obtained numerically using a cross-validation procedure by the sth cluster's variant. The coefficients α, α_g, β, β_g, and γ, γ_g are nonnegative weights chosen for the accuracy, sensitivity and specificity, respectively. They are all established by a designer (domain expert). The index "g" indicates the number of group. For some standardization, we assume that

$$\alpha + \beta + \gamma = \sum_{g=1}^{G} \left(\alpha_g + \beta_g + \gamma_g\right) = 1 \tag{14}$$

for $g = 1, \ldots, G$.

If we consider the accuracy to be the most important performance measure, then:

(a) for $G = 2$ we take $(\alpha, \beta, \gamma) = (1, 0, 0)$ and the global performance index equals $J = \mathrm{Acc}$.
(b) for $G > 2$ we take $\left(\alpha_g, \beta_g, \gamma_g\right) = \left(\frac{1}{G}, 0, 0\right)$ for $g = 1, \ldots, G$ and the global performance index is the average accuracy computed over partial accuracies separately.

The case when $J = \mathrm{Acc}$ or $J = \frac{1}{G}\sum_{g=1}^{G}\mathrm{Acc}_g$ will be called a primary one.

The k-means-based PNN reduction method is summarized in Algorithm 1.

Algorithm 1: PNN structure reduction based on k–means clustering.

```
for s := 1 to N − 1 do
    for g := 1 to G do
        Compute i_{s,g} cluster centers for training set
    end
    Perform PNN cross validation procedure on c_s = Σ_{g=1}^{G} i_{s,g} cluster centers
    Calculate global performance index J(s) as in (9)–(10) for PNN
end
Read σ_1^(1), ..., σ_n^(G) for PNN corresponding to s* with the highest value of J(s)
```

A similar solution is provided in Zaknich (1997) but that concept is dependent on various quantization levels of the input space. Here the number of clusters is determined by (7). The other difference lies in the choice of smoothing parameter. In Zaknich (1997), single σ is used for the model. In this paper, PNN adopts single σ for each variable and class. It is necessary to note that there exist a large number of other more sophisticated clustering choices which could be applied for data reduction in Algorithm 1, e.g., cluster labeling method for SV clustering (Lee and Lee 2005), spectral biclustering (Liu et al. 2006), weighted graph-based clustering (Lee and Lee 2006), parameterless clustering (Tseng and Kao 2005) or the approach which allows for large overlaps among clusters of the same class (Fu and Wang 2003). However, the use of original, basic clustering method in the present study provides satisfactory prediction results for reduced PNN; therefore, we decide to utilize this approach.

4.3 PNN structure reduction by means of support vector machines

Support vector machines (SVMs) (Vapnik 1995) are one of the most accurate methods among all well-known classification algorithms (Wu et al. 2008). SVMs construct an optimal classifier for the input vector $\mathbf{x}_i$ $(i = 1, \ldots, l)$ with associated class label $y_i = \pm 1$. Two types are usually applied in data mining problems: the C-SVM model and the ν-SVM model (Schölkopf et al. 2000). Since C-based SVM is used in this research, only the C-SVM training procedure is highlighted. In short, the C-SVM algorithm requires the solution of the following quadratic programming optimization (QP) problem

$$\begin{cases} \max\limits_{\alpha} W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, l, \quad \sum_{i=1}^{l}\alpha_i y_i = 0, \end{cases} \tag{15}$$

where α_i's are the Lagrange multipliers and $K(\cdot)$ is the kernel function. Once the solution of (15) is obtained in terms of $\boldsymbol{\alpha}$ vector, the optimal classifier is formulated

$$\mathrm{class}(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{l}\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{16}$$

The input vectors $\mathbf{x}_i$, having $\alpha_i > 0$, are called support vectors (SVs). Thus, the summation in (16) is not actually performed over all training patterns but over SVs (Kecman 2001). SVs constitute a sufficient subset out of given input data for a sample prediction. The solution of the QP problem in (15) is subject to constraint with respect to α_i,

which involves the choice of an unknown C parameter. Furthermore, the result of classification depends on the kernel function $K(\cdot)$ applied in an optimal classifier (16). Therefore, an appropriate selection of aforementioned factors is the significant issue which has to be addressed. It is done in the following subsections.

Finally, note that although the SVMs described above are binary classifiers, they are easily combined to handle multi-class classification problems. The most widely used approaches combine multiple binary classifiers trained separately using either G one-against-all (say, "one" positive, "rest" negative) or one-against-one schemes (Hsu and Lin 2002).

4.3.1 The meaning of C constraint

The coefficient C in (15) is the parameter which introduces additional capacity control for the classifier. The adjustment of C provides a greater or smaller number of support vectors which, in turn, influences the classification accuracy.

In this research, by setting different values to C constraint, we are capable of obtaining different sets of support vectors. Depending on the considered data set and the value of C, the size of PNN varies.

4.3.2 The use of kernel function

Much study in recent years has been devoted to adopting different kernels for SVM (Chapelle et al. 2002; Schölkopf and Smola 2002; Tsang et al. 2005; Chu and Wang 2005; Khandoker et al. 2009; Zhang et al. 2014). In this contribution, the Gaussian kernel function is applied with the spread constant (sc) as the parameter

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2(sc)^2}\right). \tag{17}$$

An appropriate range of the spread constant has to be estimated, which is realized numerically with the assumption of achieving the highest generalization ability of the classifier.

4.3.3 The proposed approach

For the constraint C and the spread constant sc, the final sets of values A_C and A_{sc} are assumed, respectively. The grid search method for both C and sc is performed, where $A_C = \{10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$ and $A_{sc} = \{1.2, 1.5, 2, 5, 10, 50, 80, 100, 200, 500\}$.

In general, the optimal values (C^*, sc^*) are defined as follows

$$\left(C^*, sc^*\right) = \arg\max_{(C,sc)} Q(C, sc), \tag{18}$$

where $Q(C, sc)$ is a global performance index defined as follows

$$Q(C, sc) = \begin{cases} \alpha\mathrm{Acc} + \beta\mathrm{Sen} + \gamma\mathrm{Spe} & \text{if } G = 2 \\ \sum_{g=1}^{G} \left(\alpha_g \mathrm{Acc}_g + \beta_g \mathrm{Sen}_g + \gamma_g \mathrm{Spe}_g\right) & \text{if } G > 2. \end{cases} \tag{19}$$

In contrast to (10), Acc, Sen, Spe, Acc_g, Sen_g and Spe_g depend on C and sc. The SVM-based PNN reduction methodology is summarized in Algorithm 2.

Algorithm 2: PNN structure reduction based on SVM algorithm.

```
foreach sc ∈ A_sc and C ∈ A_C do
    Perform SVM classification on whole input data set and
    select support vectors SVs
    Perform PNN cross-validation procedure using SVs as the
    input data
    Calculate global performance index Q(C, sc) as in (18)–(19)
    for PNN
end
Read σ_1^(1), ..., σ_n^(G) for PNN corresponding to C* and sc* with
the highest value of Q(C, sc)
```

In order to solve multi-class classification problems by SVM in Algorithm 2, we utilize "one-against- one" method. In this approach, for a data set with G classes, $G(G-1)/2$ binary classifiers are constructed where each one is trained using vectors from two classes. The prediction is performed on the basis of voting strategy by assigning an unknown test case to the class with the highest vote (Wang and Fu 2005).

5 Input data used to test the models

In this study, we use seven UCI machine learning repository medical data sets (Bache and Lichman 2013):

- Wisconsin breast cancer (WBC) set with 683 patterns and 9 features. The data are divided into two groups: 444 benign cases and 239 malignant cases.
- Pima Indians diabetes (PID) set with 768 patterns and 8 features. Two classes of data are considered: samples tested negative (500 women) and samples tested positive (268 women).
- Haberman's survival (HS) set with 306 patterns and 3 features. There are two input classes: patients who survived 5 years or longer (225 records) and patients who died within 5 years (81 records).

- Cardiotocography (CTG) set with 2126 patterns and 22 features. The classes are coded into three states: normal (1655 cases), suspect (295 cases) and pathological (176 cases).
- Thyroid (T) set with 7200 patterns and 21 features. Three classes are regarded: subnormal functioning (166 samples), hyperfunction (368 samples) and not hypothyroid (6666 samples).
- Dermatology (D) set with 358 patterns and 34 features. Six data classes are considered: psoriasis (111 cases), seborrheic dermatitis (60 cases), lichen planus (71 cases), chronic dermatitis (48 cases), pityriasis rosea (48 cases) and pityriasis rubra pilaris (20 cases).
- Diagnostic Wisconsin breast cancer (DWBC) set with 569 patterns and 30 features. Two medical states are regarded: malignant (212 instances) and benign (357 instances).

Additionally, the research is conducted with the use of an ovarian cancer (OC) real set of records which represent 199 women after ovarian cancer treatment. There are 17 parameters registered for each case. The 60-month survival threshold mandates two data groups: 131 cases under and 68 cases over this threshold, respectively. The data are obtained from the Clinical Department of Obstetrics and Gynecology of Rzeszow State Hospital in Poland. The analysis of ovarian cancer treatment is discussed in Skret et al. (2001).

6 Results and discussion

This section presents the comparison of the prediction ability measured for the standard PNN model and the networks for which the number of pattern neurons is reduced by means of Algorithms 1 and 2. The prediction ability of the examined classifiers is assessed using the global performance indices $J(s)$, $Q(C, sc)$ which involve the models' accuracy, sensitivity and specificity. These indices are determined on the basis of a tenfold cross-validation procedure. Furthermore, the obtained results are compared to the prediction ability of the reference classifiers: single decision tree (SDT), multilayer perceptron (MLP), SVM and k-means algorithm, and to the outcomes for the state-of-the-art PNNs training procedures. At the end of this section, we highlight some aspects related to time effectiveness of the proposed algorithms.

6.1 Results for the proposed approaches

The left-hand sides of Tables 1, 2, 3, 4, 5, 6, 7 and 8 illustrate the number of cluster centers (c_s) used in PNN structure and the performance measures Acc, Sen, Spe and J computed for PNN trained according to Algorithm 1 for each of the considered data sets. In the row with the label "All," we provide the results for PNN with all pattern neurons. The right-hand sides of these tables present the spread constant sc, the numbers of support vectors SVs, Acc, Sen, Spe and Q calculated for PNN using Algorithm 2. For multiclass classification problems, the average values of the accuracy, sensitivity and specificity are shown and denoted, respectively: $\overline{\text{Acc}}$, $\overline{\text{Sen}}$ and $\overline{\text{Spe}}$.

In our analysis, we consider two cases:

(a) the primary case when the global performance indices J and Q are the accuracies or the average accuracy values,
(b) the exemplary case, in which we assume the values of the weights for the accuracy, sensitivity and specificity according to the designer knowledge.

For two class classification problems (Tables 1, 2, 3, 7, 8), the weights are $\alpha = 0.5$, $\beta = 0.3$ and $\gamma = 0.2$ [see (10)–(14)]. In case of CTG data set classification task (Table 4), the weights are set as follows: $(\alpha_1, \beta_1, \gamma_1) = (0.2, 0.07, 0.03)$, $(\alpha_2, \beta_2, \gamma_2) = (0.05, 0.03, 0.02)$ and $(\alpha_3, \beta_3, \gamma_3) = (0.4, 0.15, 0.05)$. For T database classification problem (Table 5), the following weights are used: $(\alpha_1, \beta_1, \gamma_1) = (0.2, 0.07, 0.03)$, $(\alpha_2, \beta_2, \gamma_2) = (0.4, 0.15, 0.05)$ and $(\alpha_3, \beta_3, \gamma_3) = (0.05, 0.03, 0.02)$. Finally, for D data set classification task (Table 6), we apply: $(\alpha_g, \beta_g, \gamma_g) = (0.1, 0.0367, 0.03)$ for $g = 1, \ldots, 6$. For CTG data set, indices 1, 2 and 3 are assigned to the classes suspect, normal and pathological, respectively. In case of T database, indices 1, 2 and 3 correspond to the classes subnormal, hyperfunction and not hypothyroid, respectively. The following conclusions can be inferred:

1. In case of Algorithm 1, in seven out of eight data classification cases, by reducing the number of pattern neurons of PNN, we observe a higher value of the global performance index J than the one computed with the use of all pattern neurons of the model. The exception is in T data set classification task.
2. In five classification tasks, by reducing the number of pattern neurons of PNN by means of Algorithm 2, we obtain a higher value of the global performance index Q than the one determined for full structure network.
3. The most gainful reduction ratio R defined in (8) by optimal $s = s^*$ can be directly read from Tables 1, 2, 3, 4, 5, 6, 7 and 8 . For example, in DWBC data set classification problem, it takes the value of $R = \frac{57}{569} \approx 0.1$. Thus, instead of all the original data cases, we can use their substitutes, but about 10 times smaller in number and we get a higher value of global performance index.

It needs to be stressed that in majority of classification tasks, Algorithm 1 applied to the training process of PNN provides higher values of the global performance index J in

Table 1 Results for WBC classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^4$

Algorithm 1							Algorithm 2						
s	c_s	Acc	Sen	Spe	J	Time (s)	sc	SVs	Acc	Sen	Spe	Q	Time (s)
1	68	0.971	0.958	0.977	0.968	0.94	1.2	65	0.677	0.767	0.500	0.669	0.41
2	137	0.993	1.000	0.988	**0.994**	2.62	1.5	69	0.681	0.773	0.520	0.676	0.92
3	205	0.966	0.972	0.962	0.967	5.74	2	80	0.750	0.818	0.600	0.740	1.14
4	274	0.975	0.968	0.977	0.973	8.27	5	218	0.954	0.971	0.889	0.946	8.27
5	342	0.976	0.983	0.973	0.978	12.09	10	293	0.966	0.885	0.987	0.946	19.08
6	409	0.976	0.965	0.981	0.974	16.91	50	397	0.982	0.992	0.968	0.982	17.31
7	478	0.981	0.988	0.977	0.982	25.27	80	432	0.969	0.983	0.953	0.970	20.30
8	546	0.985	0.989	0.983	0.986	30.99	100	439	0.982	0.970	0.992	0.980	24.99
9	615	0.985	0.991	0.983	0.986	31.64	200	449	0.982	0.987	0.976	0.982	29.53
All	683	0.987	0.987	0.986	0.987	46.71	500	449	0.984	0.987	0.981	**0.984**	35.31

Table 2 Results for PID classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^2$

Algorithm 1							Algorithm 2						
s	c_s	Acc	Sen	Spe	J	Time (s)	sc	SVs	Acc	Sen	Spe	Q	Time (s)
1	77	0.909	0.852	0.940	**0.898**	2.07	1.2	374	0.486	0.259	0.709	0.463	2.49
2	154	0.759	0.611	0.840	0.731	4.45	1.5	385	0.525	0.262	0.773	0.496	3.97
3	230	0.800	0.675	0.867	0.776	9.50	2	384	0.521	0.312	0.733	0.501	7.38
4	307	0.801	0.626	0.895	0.767	15.51	5	386	0.588	0.345	0.794	0.556	12.32
5	384	0.794	0.619	0.888	0.760	19.10	10	407	0.636	0.389	0.826	0.600	13.17
6	461	0.757	0.528	0.880	0.713	19.92	50	664	0.738	0.574	0.847	0.711	31.18
7	538	0.797	0.622	0.891	0.763	39.06	80	725	0.774	0.608	0.871	0.744	38.75
8	614	0.764	0.556	0.875	0.724	40.84	100	742	0.784	0.876	0.623	**0.779**	40.15
9	691	0.769	0.573	0.876	0.732	39.79	200	768	0.778	0.608	0.870	0.745	45.54
All	768	0.778	0.608	0.870	0.745	44.02	500	768	0.778	0.608	0.870	0.745	45.51

Table 3 Results for HS classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^0$

Algorithm 1							Algorithm 2						
s	c_s	Acc	Sen	Spe	J	Time (s)	sc	SVs	Acc	Sen	Spe	Q	Time (s)
1	31	0.677	0.250	0.826	0.579	0.11	1.2	170	0.524	0.000	1.000	0.462	0.64
2	61	0.754	0.313	0.911	0.653	0.15	1.5	169	0.550	0.914	0.216	0.592	0.55
3	92	0.761	0.083	1.000	0.605	0.17	2	169	0.538	0.049	0.988	0.481	0.64
4	122	0.778	0.375	0.922	**0.686**	0.57	5	171	0.549	0.062	0.989	0.491	0.67
5	154	0.747	0.268	0.920	0.638	0.61	10	174	0.528	0.025	0.957	0.463	0.70
6	184	0.761	0.102	1.000	0.611	0.47	50	200	0.600	0.062	0.966	0.512	1.72
7	215	0.744	0.140	0.962	0.606	0.71	80	215	0.637	0.148	0.933	0.550	2.01
8	245	0.735	0.077	0.972	0.585	1.57	100	224	0.687	0.308	0.902	0.616	2.42
9	276	0.768	0.246	0.956	0.649	1.65	200	243	0.695	0.259	0.914	0.608	2.70
All	306	0.761	0.247	0.946	0.644	2.12	500	266	0.741	0.333	0.919	**0.654**	2.79

Table 4 Results for CTG classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^3$

Algorithm 1							Algorithm 2						
s	c_s	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)	sc	SVs	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	Q	Time (s)
1	214	0.981	0.920	0.979	0.955	9.26	1.2	230	0.925	0.892	0.941	0.941	12.17
2	425	0.980	0.912	0.978	0.952	31.61	1.5	247	0.933	0.897	0.944	0.945	12.78
3	639	0.982	0.916	0.974	0.957	84.11	2	288	0.928	0.886	0.942	0.933	20.92
4	850	0.981	0.920	0.975	0.957	119.93	5	600	0.971	0.941	0.972	0.969	117.65
5	1064	0.984	0.936	0.977	0.965	342.56	10	1069	0.981	0.959	0.981	0.978	215.08
6	1276	0.990	0.968	0.986	0.986	348.41	50	1985	0.991	0.977	0.989	**0.989**	605.79
7	1489	0.981	0.948	0.980	0.973	330.78	80	2070	0.979	0.917	0.979	0.954	610.80
8	1701	0.985	0.939	0.974	0.968	583.02	100	2084	0.990	0.970	0.986	0.986	777.62
9	1914	0.991	0.976	0.989	**0.988**	814.52	200	2098	0.992	0.975	0.989	0.988	745.51
All	2126	0.991	0.973	0.989	0.987	887.88	500	2110	0.992	0.978	0.991	0.989	757.76

Table 5 Results for T classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^1$

Algorithm 1							Algorithm 2						
s	c_s	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)	sc	SVs	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	Q	Time (s)
1	721	0.963	0.669	0.787	0.825	52.36	1.2	944	0.955	0.921	0.966	0.951	159.35
2	1440	0.985	0.880	0.934	0.944	112.96	1.5	943	0.947	0.913	0.959	0.944	180.56
3	2160	0.989	0.939	0.979	0.974	375.21	2	1009	0.949	0.900	0.957	0.940	202.41
4	2879	0.981	0.881	0.941	0.943	655.14	5	1187	0.954	0.913	0.964	0.948	289.59
5	3600	0.982	0.895	0.957	0.953	1380.51	10	1262	0.959	0.905	0.965	0.949	319.21
6	4321	0.982	0.891	0.950	0.950	1898.36	50	1963	0.977	0.936	0.974	0.964	635.75
7	5040	0.982	0.874	0.939	0.944	2098.01	80	2365	0.980	0.939	0.978	0.968	965.03
8	5760	0.983	0.903	0.951	0.953	2293.61	100	2598	0.981	0.929	0.973	0.964	1128.61
9	6479	0.981	0.894	0.944	0.947	4338.36	200	3449	0.987	0.951	0.977	0.974	1500.32
All	7200	0.994	0.963	0.985	**0.985**	7543.13	500	5021	0.991	0.960	0.980	**0.980**	2833.95

Table 6 Results for D classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^3$

Algorithm 1							Algorithm 2						
s	c_s	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)	sc	SVs	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	Q	Time (s)
1	36	0.991	0.917	0.995	0.975	0.35	1.2	257	0.999	0.996	0.999	0.998	23.34
2	72	0.999	0.999	0.999	**0.999**	0.67	1.5	282	0.998	0.991	0.998	0.997	24.24
3	106	0.991	0.967	0.994	0.986	2.82	2	319	0.999	0.997	0.999	**0.999**	29.96
4	142	0.995	0.985	0.997	0.994	3.23	5	356	0.999	0.997	1.000	0.999	33.31
5	180	0.998	0.993	0.999	0.997	7.11	10	358	0.997	0.990	0.999	0.996	31.86
6	216	0.995	0.978	0.997	0.992	10.34	50	358	0.997	0.990	0.999	0.996	31.90
7	252	0.999	0.996	0.999	0.998	13.64	80	358	0.997	0.990	0.999	0.996	32.28
8	286	0.999	0.996	0.999	0.998	18.08	100	358	0.997	0.990	0.999	0.996	32.10
9	322	0.997	0.990	0.998	0.996	24.99	200	358	0.997	0.990	0.999	0.996	32.13
All	358	0.997	0.990	0.998	0.996	34.88	500	358	0.997	0.990	0.999	0.996	31.79

Table 7 Results for DWBC classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^{-1}$

Algorithm 1							Algorithm 2						
s	c_s	Acc	Sen	Spe	J	Time (s)	sc	SVs	Acc	Sen	Spe	Q	Time (s)
1	57	0.999	0.997	0.998	**0.998**	0.62	1.2	206	0.976	0.980	0.971	0.976	56.48
2	113	0.991	1.000	0.976	0.991	2.67	1.5	211	0.976	0.963	0.990	0.975	62.20
3	171	0.998	0.997	0.997	0.997	7.48	2	226	0.982	1.000	0.965	0.984	74.32
4	228	0.997	0.995	0.994	0.996	15.78	5	335	0.976	0.976	0.976	0.976	107.65
5	285	0.986	0.983	0.991	0.986	22.41	10	442	0.989	0.981	0.996	0.988	132.06
6	341	0.982	0.995	0.961	0.982	30.78	50	568	0.993	0.986	0.997	**0.992**	183.98
7	398	0.992	0.996	0.986	0.992	50.52	80	569	0.993	0.986	0.997	0.992	189.27
8	456	0.991	0.993	0.988	0.991	92.91	100	569	0.993	0.986	0.997	0.992	189.27
9	512	0.988	0.997	0.974	0.988	104.80	200	569	0.993	0.986	0.997	0.992	189.27
All	569	0.993	0.997	0.986	0.993	181.80	500	569	0.993	0.986	0.997	0.992	189.27

Table 8 Results for OC classification task: Algorithm 1—the number of cluster centers (c_s); Algorithm 2—the spread constant and the number of support vectors for $C^* = 10^{-1}$

Algorithm 1							Algorithm 2						
s	c_s	Acc	Sen	Spe	J	Time (s)	sc	SVs	Acc	Sen	Spe	Q	Time (s)
1	20	0.650	0.143	0.923	0.553	0.16	1.2	155	0.826	0.867	0.793	0.832	5.16
2	40	0.990	1.000	0.985	**0.992**	0.47	1.5	163	0.883	0.882	0.884	0.883	6.52
3	59	0.966	0.950	0.974	0.963	1.61	2	171	0.859	0.882	0.845	0.863	6.07
4	79	0.975	0.963	0.981	0.973	2.28	5	183	0.853	0.794	0.887	0.842	5.92
5	100	0.990	1.000	0.985	0.992	3.23	10	189	0.867	0.853	0.876	0.865	6.09
6	120	0.933	0.902	0.949	0.927	3.48	50	195	0.892	0.867	0.905	**0.887**	9.86
7	140	0.914	0.854	0.946	0.902	4.76	80	196	0.857	0.838	0.867	0.853	10.88
8	159	0.906	0.907	0.905	0.906	9.42	100	196	0.857	0.838	0.867	0.853	11.18
9	179	0.916	0.902	0.924	0.913	9.68	200	196	0.857	0.838	0.867	0.853	11.18
All	199	0.864	0.867	0.863	0.865	11.89	500	197	0.878	0.824	0.907	0.868	9.91

Table 9 The quotient of the original PNN training time to the training time of reduced PNN

Data set	Algorithm 1	Algorithm 2
WBC	17.83	⋆
PID	21.26	1.13
HS	3.72	⋆
CTG	1.09	1.25
T	⋆	⋆
D	52.06	1.06
DWBC	293.23	1.03
OC	25.29	1.01

comparison with Q obtained by PNN trained by means of Algorithm 2.

Table 9 contains the quotient of the original PNN training time to the training time of reduced PNN, only if the Algorithm i ($i = 1, 2$) provides the highest value of the global performance index (J or Q, respectively).

The symbol "⋆" means that the global performance index of Algorithm i, ($i = 1, 2$) is lower than the one provided by original PNN. We observe that in the case of Algorithm 1, this quotient is greater than 1 in seven out of eight data classification tasks taking the highest value of 293.23 (DWBC problem). However, as it can be seen from Table 9, Algorithm 2 is much more time-consuming in comparison with Algorithm 1.

6.2 Comparison to reference classifiers

All the reference classifiers utilized in the classification problems are trained and tested in DTREG software (Sherrod 2015). Below, the short description of the model's settings is highlighted.

SDT is simulated with the entropy to evaluate the quality of splits in the process of tree construction. The depth of the tree is set to 10. The pruning algorithm is applied to find the optimal tree size. We prune the tree with respect to minimum cross-validation error.

MLP is trained with one or two hidden layers. Linear or logistic transfer functions are used for activation of the neurons in hidden and output layers. The search for the optimal number of hidden layer neurons is performed in order to minimize the sum squared error of model. The scaled conjugate gradient algorithm is MLP's training algorithm.

SVM algorithm is also applied in this research as the reference classifier. Multiclass classification tasks are solved using the one-versus-one approach. In each classification problem, radial basis kernel function is utilized with experimental grid search for model's parameters C and sc.

Similarly to SVM, the k-means clustering algorithm is the reference model applied in this work for comparison purposes. The k-means predictions for the unknown patterns are determined by using the category of the nearest cluster. In the experiments, we search for the number of clusters for which the highest testing accuracy is obtained.

Tables 10, 11, 12 and 13 present the performance measures Acc, Sen, Spe and J computed for MLP, SDT, SVM and k-means algorithms in all considered classification problems.

We can observe that in six out of eight data set classification tasks, PNN with the structure reduced by means of Algorithm 1 or Algorithm 2 yields a higher value of the global performance index.

6.3 Comparison to state-of-the-art procedures

In this section, the classification accuracy values obtained by two proposed approaches are compared to the accuracies for PNNs available in the literature. Table 14 shows the results for WBC, PID, HS, T, D and DWBC data set classification tasks.

In this table, for comparison purposes, we also present the accuracy values provided by PNN with all hidden neurons in the pattern layer and the ones for the reference classifiers. The best results are marked with bold. As shown, in each data classification case, the PNN models trained by means of Algorithm 1 or Algorithm 2 outperform PNNs trained using state-of-the-art methods (Georgiou et al. 2006; Chang et al. 2008; Georgiou et al. 2008; Saiti et al. 2009; Temurtas et al. 2009; Chandra and Babu 2011; Yeh and Lin 2011; Azar and El-Said 2013). Our algorithms also perform better than the reference classifiers in all considered data set classification problems. Only in for T data classification task, PNN with all neurons in the pattern layer yields the highest value of the

Table 10 Results for the reference classifiers in the classification tasks of WBC and PID data sets

Classifier	WBC					PID				
	Acc	Sen	Spe	J	Time (s)	Acc	Sen	Spe	J	Time (s)
MLP	0.968	0.949	0.977	0.964	3.94	0.769	0.578	0.872	**0.732**	3.81
SDT	0.950	0.928	0.962	0.946	0.22	0.748	0.608	0.824	0.721	0.39
SVM	0.972	0.979	0.968	**0.973**	11.33	0.772	0.548	0.892	0.729	291.98
k-means	0.956	0.925	0.973	0.950	335.16	0.691	0.425	0.834	0.640	1.74

Table 11 Results for the reference classifiers in the classification tasks of HS and CTG data sets

Classifier	HS					CTG				
	Acc	Sen	Spe	J	Time (s)	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)
MLP	0.728	0.161	0.933	0.599	2.78	0.985	0.949	0.980	0.974	77.31
SDT	0.748	0.395	0.875	**0.668**	0.21	0.991	0.977	0.986	**0.990**	0.38
SVM	0.742	0.111	0.968	0.598	233.36	0.987	0.951	0.982	0.976	157.32
k-Means	0.686	0.383	0.796	0.617	8.96	0.936	0.842	0.926	0.919	763.34

Table 12 Results for the reference classifiers in the classification tasks of T and D data sets

Classifier	T					D				
	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)	$\overline{\text{Acc}}$	$\overline{\text{Sen}}$	$\overline{\text{Spe}}$	J	Time (s)
MLP	0.966	0.645	0.806	0.825	146.03	0.988	0.963	0.993	0.984	5.16
SDT	0.990	0.949	0.977	**0.977**	0.50	0.980	0.914	0.988	0.967	0.38
SVM	0.986	0.868	0.936	0.945	451.57	0.991	0.969	0.994	**0.987**	12.50
k-Means	0.895	0.634	0.797	0.794	122,040.77	0.966	0.885	0.980	0.951	410.75

Table 13 Results for the reference classifiers in the classification tasks of DWBC and OC data sets

Classifier	DWBC					OC				
	Acc	Sen	Spe	J	Time (s)	Acc	Sen	Spe	J	Time (s)
MLP	0.975	0.957	0.986	**0.972**	6.81	0.814	0.808	0.817	0.813	2.41
SDT	0.936	0.896	0.961	0.929	0.54	0.758	0.750	0.763	0.757	0.25
SVM	0.975	0.958	0.986	**0.972**	6.11	0.849	0.808	0.870	**0.841**	4.97
k-means	0.891	0.778	0.958	0.871	98.17	0.758	0.779	0.748	0.762	0.59

Table 14 The accuracy results in the classification of WBC, PID, HS, T, D and DWBC data sets for the proposed approaches, PNN trained with total number of pattern neurons, the reference classifiers and the state-of-the-art PNN learning algorithms

Data set	Proposed approaches		Full PNN	Reference classifiers				State-of-the-art methods	
	Algorithm 1	Algorithm 2		MLP	SDT	SVM	k-Means	Source	Result
WBC	**0.993**	0.984	0.987	0.968	0.950	0.972	0.956	Georgiou et al. (2008)	0.989
								Azar and El-Said (2013)	0.976
PID	**0.909**	0.784	0.778	0.769	0.748	0.772	0.691	Temurtas et al. (2009)	0.781
								Georgiou et al. (2006)	0.753
HS	**0.778**	0.741	0.761	0.728	0.748	0.742	0.686	Chandra and Babu (2011)	0.743
T	0.989	0.991	**0.994**	0.966	0.990	0.986	0.895	Yeh and Lin (2011)	0.983
								Saiti et al. (2009)	0.968
D	**0.999**	**0.999**	0.997	0.988	0.980	0.991	0.966	Chang et al. (2008)	0.935
DWBC	**0.999**	0.993	0.993	0.975	0.936	0.975	0.891	Chang et al. (2008)	0.954

accuracy. However, our result is worse only by a margin of 0.3 %.

In our paper, in both Algorithms 1 and 2, the smoothing parameters are determined experimentally in a way that the global performance indices achieve maximal value. The authors of Xu et al. (1994) provided a theorem on the selection of this parameter for designing Parzen window estimator particularly for probabilistic neural network. It seems advisable utilizing this interesting result in future.

7 Conclusions

This article constituted the generalization of the previous authors' results in Kusy and Kluska (2013). In the current study, we conducted more comprehensive analysis on the problem of PNN structure reduction. Firstly, the prediction ability of reduced PNN was assessed by means of a tenfold cross-validation procedure. Such an approach is commonly used for algorithms testing purposes. Secondly, we proposed the global performance index, which included the accuracy, sensitivity and specificity in order to determine the prediction ability of the considered models. The global performance index presented in this way is quite flexible since it can take the form of model's accuracy or the form of weighted accuracy, sensitivity and specificity values for each class separately. The values of particular weights can be established by a designer (domain expert) according to his/her best knowledge. We would like to stress that this is particularly important especially in medical data classification problems, as the ones used in this study. Furthermore, the PNN classifiers with the number of pattern neurons reduced by means of k-means clustering and SVM procedure were compared to well-known computational intelligence algorithms: single decision tree, multilayer perceptron, support vector machines and k-means clustering procedure. In six classification tasks, we achieved a higher value of the global performance index for PNN with reduced architecture than for the considered reference classifiers. Finally, we also made the comparison of the accuracy values of the reduced PNN models and PNNs trained by state-of-the-art procedures. In all data classification cases, the accuracies obtained by means of our algorithms took a higher value.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Human and animal participants Research is not involved with human participants and/or animals.

Ethical standards Seven data sets used in this work are taken from UCI machine learning repository. There is also a single database obtained from the Clinical Department of Obstetrics and Gynecology of Rzeszow State Hospital in Poland.

References

Adeli H, Panakkat A (2009) A probabilistic neural network for earthquake magnitude prediction. Neural Netw 22(7):1018–1024

Altman DG, Bland JM (1994) Diagnostic tests 1: sensitivity and specificity. Br Med J 308(6943):1552

Azar AT, El-Said SA (2013) Probabilistic neural network for breast cancer classification. Neural Comput Appl 23(6):1737–1751

Bache K, Lichman M (2013) UCI machine learning repository, Technical Report. School Information and Computer Science, University of California, Irvine. http://archive.ics.uci.edu/ml. Accessed 15 Oct 2015

Berthold MR, Diamond J (1998) Constructive training of probabilistic neural networks. Neurocomputing 19(1–3):167–183

Burrascano P (1991) Learning vector quantization for the probabilistic neural network. IEEE Trans Neural Netw 2(4):458–461

Chandra B, Babu KVN (2011) An improved architecture for probabilistic neural networks. In: Proceedings of IEEE international joint conference on neural networks, pp 919–924

Chang RKY, Loo CK, Rao MVC (2008) A global k-means approach for autonomous cluster initialization of probabilistic neural network. Informatica 32(2):219–225

Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for support vector machines. Mach Learn 46(1–3):131–159

Chtioui Y, Bertrand D, Barba D (1996) Reduction of the size of the learning data in a probabilistic neural network by hierarchical clustering. Application to the discrimination of seeds by artificial vision. Chemom Intell Lab Syst 35(2):175–186

Chtioui Y, Panigrahi S, Marsh R (1998) Conjugate gradient and approximate Newton methods for an optimal probabilistic neural network for food color classification. Opt Eng 37(11):3015–3023

Chu F, Wang LP (2005) Applications of support vector machines to cancer classification with microarray data. Int J Neural Syst 15(6):475–484

Folland R, Hines E, Dutta R, Boilot P, Morgan D (2004) Comparison of neural network predictors in the classification of tracheal-bronchial breath sounds by respiratory auscultation. Artif Intell Med 31(3):211–220

Fu XJ, Wang LP (2003) Data dimensionality reduction with application to simplifying RBF network structure and improving classification performance. IEEE Trans Syst Man Cybern Part B Cybern 33(3):399–409

Georgiou LV, Pavlidis NG, Parsopoulos KE, Alevizos PD, Vrahatis MN (2006) New self-adaptive probabilistic neural networks in bioinformatic and medical tasks. Int J Artif Intell Tools 15(3):371–396

Georgiou VL, Alevizos PD, Vrahatis MN (2008) Novel approaches to probabilistic neural networks through bagging and evolutionary estimating of prior probabilities. Neural Process Lett 27(2):153–162

Gorunescu F, Gorunescu M, El-Darzi E, Gorunescu S (2005) An evolutionary computational approach to probabilistic neural network with application to hepatic cancer diagnosis. In: Tsymbal A, Cunningham P (eds) Proceedings of IEEE symposium on computer-based medical systems. IEEE Computer Society Press, Los Alamitos, pp 461–466

Hartigan JA, Wong MA (1979) A k-means clustering algorithm. J R Stat Soc Ser C (Appl Stat) 28(1):100–108

Hsu C-W, Lin C-J (2002) A comparison of methods for multi-class support vector machines. IEEE Trans Neural Netw 13(2):415–425

Huang C-J, Liao W-C (2004) Application of probabilistic neural networks to the class prediction of leukemia and embryonal tumor of central nervous system. Neural Process Lett 19(3):211–226

Kecman V (2001) Learning and soft computing, support vector machines, neural networks and fuzzy logic models. The MIT Press, Cambridge

Khandoker AH, Palaniswami M, Karmakar CK (2009) Support vector machines for automated recognition of obstructive sleep apnea syndrome from ECG recordings. IEEE Trans Inf Technol Biomed 13(1):37–48

Kluska J (2009) Analytical methods in fuzzy modeling and control. Springer, Berlin

Kowalski PA, Kulczycki P (2014) Neural classification for interval information. In: Agre G et al (eds) Lecture notes in computer science, vol 8722. Springer, Berlin, pp 206–213

Kusy M, Kluska J (2013) Probabilistic neural network structure reduction for medical data classification. In: Rutkowski L, Korytkowski M, Sherer R, Tadeusiewicz R, Zadeh LA, Zurada J (eds) Lecture notes in artificial intelligence Part I. Springer, Berlin, pp 118–129

Kusy M, Zajdel R (2015) Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network. IEEE Trans Neural Netw Learn Syst 26(9):2163–2175

Lee J, Lee D (2005) An improved cluster labeling method for support vector clustering. IEEE Trans Pattern Anal Mach Intell 27(3):461–464

Lee J, Lee D (2006) Dynamic characterization of cluster structures for robust and inductive support vector clustering. IEEE Trans Pattern Anal Mach Intell 28(11):1869–1874

Liu B, Wan C, Wang LP (2006) An efficient semi-unsupervised gene selection method via spectral biclustering. IEEE Trans Nanobiosci 5(2):110–114

Lloyd SP (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

Mantzaris D, Anastassopoulos G, Adamopoulos A (2011) Genetic algorithm pruning of probabilistic neural networks in medical disease estimation. Neural Netw 24(8):831–835

Mao KZ, Tan K-C, Ser W (2000) Probabilistic neural-network structure determination for pattern classification. IEEE Trans Neural Netw 11(4):1009–1016

Masters T (1993) Practical neural networks recipes in C++. Academic Press, San Diego

Masters T (1995) Advanced algorithms for neural networks. Wiley, New York

Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 33(3):1065–1076

Ramakrishnan S, Selvan S (2007) Image texture classification using wavelet based curve fitting and probabilistic neural network. Int J Imaging Syst Technol 17(4):266–275

Riordan J (1958) Introduction to combinatorial analysis. Wiley, New York

Saiti F, Naini AA, Shoorehdeli MA, Teshnehlab M (2009) Thyroid disease diagnosis based on genetic algorithms using PNN and SVM. In: Proceedings of IEEE international conference on bioinformatics and biomedical engineering, pp 1–4

Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge

Schölkopf B, Smola AJ, Williamson RC, Bartlett PL (2000) New support vector algorithms. Neural Comput 12(5):1207–1245

Shan Y, Zhao R, Xu G, Liebich HM, Zhang Y (2002) Application of probabilistic neural network in the clinical diagnosis of cancers based on clinical chemistry data. Anal Chim Acta 471(1):77–86

Sherrod PH (2015) DTREG predictive modelling software http://www.dtreg.com. Accessed 15 Oct 2015

Skret A, Lozinski T. Chrusciel A (2001) Epidemiology of ovarian cancer: hormonal and genetic aspects. In: CIC international edition, Rome, pp 189–205

Specht DF (1990) Probabilistic neural networks and the polynomial adaline as complementary techniques for classification. IEEE Trans Neural Netw 1(1):111–121

Specht DF (1992) Enhancements to the probabilistic neural networks. In: Proceedings of IEEE international joint conference on neural networks, vol 1, Baltimore, pp 761–768

Specht DF, Romsdahl H (1994) Experience with adaptive probabilistic neural networks and adaptive general regression neural networks. In: Proceedings of IEEE world congress on computational intelligence, vol 2, Orlando, pp 1203–1208

Streit RL, Luginbuhl TE (1994) Maximum likelihood training of probabilistic neural networks. IEEE Trans Neural Netw 5(5):764–783

Temurtas H, Yumusak N, Temurtas F (2009) A comparative study on diabetes disease diagnosis using neural networks. Expert Syst Appl 36(4):8610–8615

Traven HGC (1991) A neural network approach to statistical pattern classification by 'semiparametric' estimation of probability density functions. IEEE Trans Neural Netw 2(3):366–377

Tsang IW, Kwok JT, Cheung P-M (2005) Core vector machines: fast SVM training on very large data sets. J Mach Learn Res 6:363–392

Tseng VS, Kao C-P (2005) Efficiently mining gene expression data via a novel parameterless clustering method. IEEE/ACM Trans Comput Biol Bioinf 2:355–365

Vapnik V (1995) The nature of statistical learning theory. Springer, New York

Venkatesh S, Gopal S (2011) Orthogonal least square center selection technique–a robust scheme for multiple source partial discharge pattern recognition using radial basis probabilistic neural network. Expert Syst Appl 38(7):8978–8989

Wang LP, Fu XJ (2005) Data mining with computational intelligence. Springer, Berlin

Wen X-B, Zhang H, Xu X-Q, Quan J-J (2008) A new watermarking approach based on probabilistic neural network in wavelet domain. Soft Comput 13(4):355–360

Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H et al (2008) Top 10 algorithms in data mining. Knowl Inf Syst 14(1):1–37

Xu L, Krzyzak A, Yuille A (1994) On radial basis function nets and kernel regression: statistical consistency, convergence rates and receptive field size. Neural Netw 7(4):609–628

Yeh I-C, Lin K-C (2011) Supervised learning probabilistic neural networks. Neural Process Lett 34(2):193–208

Zaknich A (1997) A vector quantisation reduction method for the probabilistic neural network. In: Proceedings of IEEE international conference on neural networks, Huston, pp 1117–1120

Zhang L, Wang LP, Lin W, Yan S (2014) Geometric optimum experimental design for collaborative image retrieval. IEEE Trans Circuits Syst Video Technol 24(2):346–359

Zhong M, Coggeshall D, Ghaneie E, Pope T, Rivera M, Georgiopoulos M et al (2007) Gap-based estimation: choosing the smoothing parameters for probabilistic and general regression neural networks. Neural Comput 19(10):2840–2864