

Fast Projection Pursuit Based on Quality of Projected Clusters

Marek Grochowski¹ and Włodzisław Duch^{1,2}

¹ Department of Informatics, Nicolaus Copernicus University, Toruń, Poland

² School of Computer Engineering, Nanyang Technological University, Singapore

grochu@is.umk.pl

Google: W. Duch

Abstract. Projection pursuit index measuring quality of projected clusters (QPC) introduced recently optimizes projection directions by minimizing leave-one-out error searching for pure localized clusters. QPC index has been used in constructive neural networks to discover non-local clusters in high-dimensional multi-class data, reduce dimensionality, aggregate features, visualize and classify data. However, for n training instances such optimization requires $O(n^2)$ calculations. Fast approximate version of QPC introduced here obtains results of similar quality with $O(n)$ effort, as illustrated in a number of classification and data visualization problems.

Keywords: Projection pursuit, Classification, Dimensionality reduction, Naive Bayes, Neural networks.

1 Introduction

Projection pursuit (PP) searches for the most “interesting” projections of multidimensional data by optimizing some objective functions referred to as the projection index [1,2]. Many projection indices have been introduced, both for unsupervised and for supervised learning. Such algorithms as the principal component analysis (PCA), independent component analysis (ICA) and Fisher’s discriminant analysis (FDA) are special cases of the projection pursuit approach. The “pursuit” aspect involves search for sequence of unique projections that gives different, low-dimensional insight into data structures. Most PP algorithms, including the QPC presented here, use linear projections.

The major advantage of PP methods is the potential to avoid the “curse of dimensionality” by reducing the data to low-dimensional space. The noisy and non-informative features are ignored, and only most valuable relations, depending on the definition of projection index, are preserved in the reduced space. Transformation of data by PP may be presented in form of a feedforward neural network where sequence of hidden nodes represent successive projections and learning procedure corresponds to optimization of certain PP index. In contrast to the backpropagation learning of all network parameters at the same time PP indices define intermediate goals for learning [3] making the final separation of the data much easier. The Quality of Projected Clusters (QPC) projection

index [4] is aimed at finding linear transformations that create compact clusters of vectors, each with vectors from a single class, separated from other clusters. Each QPC node may thus map data into several useful pure clusters, while sigmoid functions in multi-layer perceptrons (MLPs) perform much simpler mappings. In the next section the QPC index is described and some modifications that decrease computational cost are discussed, followed by comparison of speed of learning and quality of generated projections in terms of classification generalization. Various algorithms may be used in the space generated by QPC transformations. Here we have used Naive Bayes algorithm on the trained data, and on the same data reduced by the fast and the original QPC version.

2 The QPC Projection Index Learning Speed Improvement

For a given dataset $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset \mathcal{R}^d$, where each vector \mathbf{x}_i is associated with class $\mathcal{C}(\mathbf{x}_i)$, the QPC index is defined by [5]:

$$QPC(\mathbf{w}) = \sum_{i,j=1}^n \alpha_{ij} G(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)) \quad (1)$$

where α_{ij} are real constants that satisfy conditions: if $\mathcal{C}(\mathbf{x}_i) = \mathcal{C}(\mathbf{x}_j)$ then $\alpha_{ij} > 0$ and if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}(\mathbf{x}_j)$ then $\alpha_{ij} < 0$. Function $G(x)$ should be localized with maximum for $x = 0$, for example it is a Gaussian function. Then for a given direction $\mathbf{w} \in \mathcal{R}^d$ vectors \mathbf{x}_i and \mathbf{x}_j will increase QPC value if after projection on \mathbf{w} they fall close to each other and are from the same class, but if they are from different classes QPC index is decreased by a value dependent on distance between these vectors after projection on \mathbf{w} . Thus maximization of Eq. (1) leads to linear transformation that create compact and pure clusters of vectors from the same class, well separated from other clusters, provides a leave-one-out estimator measuring quality of this projection. Proper choice of constants α_{ij} and width of function $G(x)$ might force QPC optimization to prefer solutions with higher between-cluster separation over solutions characterized with better within-class purity and compactness. In all experiments presented in this paper Gaussian functions were used for localization. To normalize the QPC index value $\alpha_{ij} = 1/nn_j$ is used for all $i, j = 1, \dots, n$ satisfying condition $\mathcal{C}(\mathbf{x}_i) = \mathcal{C}(\mathbf{x}_j)$ and $\alpha_{ij} = 1/n(n - n_j)$ if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}(\mathbf{x}_j)$, where n_j denote number of instances that belong to class associated with \mathbf{x}_j , and n is the number of all instances.

Optimization of the QPC index provides solutions that might be useful in many machine learning supervised learning applications for data visualization and dimensionality reduction. Recently [5,6] this index was successfully applied to train and construct several neural networks architectures for classifications of multi-class problems.

Major disadvantage of QPC (like most of the projection pursuit indexes) is high computational cost. Each evaluation of Eq. (1) has computational complexity $O(dn^2)$, where d is the number of dimensions and n is the number of instances in training dataset, which may make this approach useless for datasets with large number of instances, especially when many iterations is needed for convergence of the optimization process. This drawback can be overcome by using a set of prototypes $\mathcal{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_k\}$ as a reference points providing estimation for dataset class distribution. For given set of

prototypes \mathcal{T} , where each prototype \mathbf{t}_i is associated with class $\mathcal{C}(\mathbf{t}_i)$, the approximation of the QPC index might be expressed as follows:

$$QPC(\mathbf{w}) = \sum_{j=1}^k \sum_{i=1}^n \alpha_{ij} G(\mathbf{w}^T(\mathbf{x}_i - \mathbf{t}_j)) \quad (2)$$

where constants $\alpha_{ij} > 0$ if $\mathcal{C}(\mathbf{x}_i) = \mathcal{C}(\mathbf{t}_j)$ and $\alpha_{ij} < 0$ if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}(\mathbf{t}_j)$, accordingly. If positions of prototypes are not fixed then Eq. (2) has $(k+1) \times d$ parameters to optimize (where k is the number of prototypes) while optimization of Eq. (1) must adjust only d weight components. However, if $k \ll n$ then computational cost becomes linear in the number of instances and in the number of features $O(kdn)$.

Solutions generated by maximization of Eq. (2) strongly depend on the number of prototypes and their initialization (position and label association). The algorithm described below allows for computing an approximation to the QPC index value for a given direction without the need of finding reference points, and might also be used for estimation of initial positions of prototypes.

Consider the set of vectors $\mathbf{x}_i \in \mathcal{R}^d$ ($i = 1, \dots, n$) projected on the \mathbf{w} direction, with the whole span of projected points divided into k equal intervals of width h :

$$y_{min} = \min_i \mathbf{w}^T \mathbf{x}_i, \quad y_{max} = \max_i \mathbf{w}^T \mathbf{x}_i, \quad h = \frac{1}{k} (y_{max} - y_{min}). \quad (3)$$

Let β_i be the center of the i -th interval:

$$\beta_i = y_{min} + h(i - 1/2), \quad i = 1, \dots, k. \quad (4)$$

For each class \mathcal{C}_i and j -th interval the partial QPC index is defined by:

$$\tilde{Q}_{\mathcal{C}_i, j} = \sum_{i=1}^n \alpha_{ij} G(\mathbf{w}^T \mathbf{x}_i - \beta_j) \quad (5)$$

where $\alpha_{ij} > 0$ if $\mathcal{C}(\mathbf{x}_i) = \mathcal{C}_j$ and $\alpha_{ij} < 0$ if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}_j$. Let associate interval j with class \mathcal{C}_i that gives maximum:

$$\mathcal{C}(\beta_j) = \arg \max_{\mathcal{C}_i} \tilde{Q}_{\mathcal{C}_i, j} \quad (6)$$

The approximate value of QPC index for direction \mathbf{w} and k intervals is computed from:

$$QPC(\mathbf{w}) \approx \sum_{j=1}^k \sum_{i=1}^n \alpha_{ij} G(\mathbf{w}^T \mathbf{x}_i - \beta_j) \quad (7)$$

where $\alpha_{ij} > 0$ if $\mathcal{C}(\mathbf{x}_i) = \mathcal{C}(\beta_j)$ and $\alpha_{ij} < 0$ if $\mathcal{C}(\mathbf{x}_i) \neq \mathcal{C}(\beta_j)$. The computational cost of evaluation of Eq. (7) is $O(kndc)$ where c denotes the number of classes. Eq. (7) might be directly used for searching for optimal \mathbf{w} , however this approximation is used here only for setting initial positions of the prototypes and their labels. Direction \mathbf{w} define line in d dimensional space $\mathbf{y} = \gamma \mathbf{w} + \boldsymbol{\mu}$, where $\gamma \in \mathcal{R}$ and $\boldsymbol{\mu} \in \mathcal{R}^d$ is an arbitrary point along this line that may be taken as the center position of all data vectors

\mathcal{X} . Then for a given direction \mathbf{w} and k intervals with centers in β_i , initial positions of prototypes $\mathbf{t}_i \in \mathcal{R}^d$ placed on this line are given by:

$$\mathbf{t}_i = \beta_i \mathbf{w} + (\boldsymbol{\mu} - (\mathbf{w}^T \boldsymbol{\mu}) \mathbf{w}). \quad (8)$$

These prototypes are used here to initialize optimization procedure of the QPC index given by Eq. (2).

Maximum number of prototypes do not exceed the number of intervals k , but might be reduced if prototypes for the same class become neighbors after projection. Additionally, the width of these intervals give a direct estimation of the spread of $G(x)$ function. For Gaussian functions setting the standard deviation to $\sigma = h$ guarantees that the partial QPC function $\hat{Q}_{c_i,j}$ given by Eq. (5), will depend mostly on data projected inside the i -th interval, and to a lesser extent on vectors that belong to the adjacent intervals.

3 Results

3.1 Learning Speed Comparison

Tab. 1 presents comparison of time needed for training of the standard QPC index defined by Eq. 1 (denoted here as QPC1) and the approximated QPC index (denoted here as QPC2) defined by Eq. (2) for several classification problems with various size and complexity of inherent relations. Most of these datasets come from the UCI repository [7] (Abalone, Appendicitis, Australian Credit Rating, Breast Cancer Wisconsin, Glass, Heart, Ionosphere, Iris, Ljubljana Breast Cancer, Monk's 1 training part, Congressional Voting Records, Spam and Wine). In addition two artificial dataset were used: 10-dimensional parity problem and Concentric Rings dataset containing 2 important features defining points inside 4 rings (one per class) and 2 noise variables drawn from uniform distribution. Both QPC1 and QPC2 use Gaussian function for $G(x)$ and a gradient descent procedure with the same learning rate (0.1) and the same stop condition. Initial positions of the prototypes for QPC2 have been set according to Eq. (8) with number of intervals $k = 20$. To avoid occurrence of local minima each optimization process was initialized 10 times with different weight values \mathbf{w} between $[-0.5, 0.5]$ and after short optimization the most promising solution has been converged to the final value. Each learning procedure was repeated 10 times and the average time required for convergence, the number of iterations and the final index value are reported in Tab. 1. Value of projection index referred in Tab. 1, both for QPC1 and QPC2, have been computed according to Eq. (1).

Results presented in Tab. 1 show great improvement of QPC2 performance compared to the QPC1. The Wilcoxon's signed-rank test [8] indicates significant difference of the average time used for computation at a confidence level of 99% (p -value of 0.0061) in favor of QPC2. Reduction of computation time occurs especially for the datasets with large number of instances like Abalone and Spam. Results for those data were excluded from statistical analysis to avoid dominance of these large values.

Projections obtained from QPC2 provide good approximation of solutions that might be found by the full QPC1 index. In most cases improvement of performance involves only slight loss of quality of obtained solutions. Fig. 1 presents scatter plots generated

Table 1. Comparison of performance of the full (QPC1) and approximate optimization (QPC2) of the QPC index

| Data Set | Vec. | Feat. | Class | QPC1 | | | QPC2 | | |
|---------------------|------|-------|-------|---------------------------|--------------------|-------------------|---------------------------|-----------------|-------------------|
| | | | | Index $\times 10^{-2}$ | Time [s] | Iterations | Index $\times 10^{-2}$ | Time [s] | Iterations |
| Appendicitis | 106 | 7 | 2 | 35.5 \pm 0.2 | 3.6 \pm 0.9 | 163.0 \pm 95.5 | 32.3 \pm 0.5 | 4.3 \pm 0.6 | 111.0 \pm 47.0 |
| Monk's 1 | 124 | 6 | 2 | 15.2 \pm 0.9 | 3.7 \pm 1.6 | 148.0 \pm 71.7 | 12.2 \pm 1.4 | 3.9 \pm 0.4 | 101.0 \pm 34.5 |
| Iris | 150 | 4 | 3 | 76.5 \pm 0.1 | 2.0 \pm 0.3 | 46.5 \pm 12.3 | 75.6 \pm 0.5 | 2.4 \pm 0.1 | 58.0 \pm 13.8 |
| Wine | 178 | 13 | 3 | 64.9 \pm 0.0 | 3.7 \pm 0.4 | 77.0 \pm 4.2 | 61.8 \pm 0.6 | 4.0 \pm 0.1 | 109.5 \pm 15.2 |
| Ionosphere | 200 | 34 | 2 | 47.1 \pm 0.2 | 16.6 \pm 11.7 | 213.0 \pm 77.3 | 41.6 \pm 0.9 | 5.0 \pm 0.2 | 110.0 \pm 24.7 |
| Sonar | 208 | 60 | 2 | 37.3 \pm 0.4 | 27.4 \pm 19.7 | 178.0 \pm 20.3 | 32.0 \pm 0.5 | 7.8 \pm 0.1 | 144.0 \pm 10.7 |
| Glass | 214 | 9 | 6 | 31.2 \pm 0.0 | 5.0 \pm 0.6 | 84.5 \pm 19.9 | 28.3 \pm 1.5 | 5.0 \pm 0.5 | 117.0 \pm 30.0 |
| Heart Statlog | 270 | 13 | 2 | 29.9 \pm 0.3 | 20.3 \pm 1.9 | 238.0 \pm 44.3 | 28.3 \pm 0.5 | 6.8 \pm 0.7 | 170.5 \pm 47.4 |
| L.Breast | 277 | 9 | 2 | 13.5 \pm 0.1 | 14.5 \pm 3.9 | 217.5 \pm 111.5 | 10.6 \pm 1.3 | 6.7 \pm 0.8 | 107.5 \pm 54.8 |
| Heart Cleveland | 297 | 13 | 2 | 29.4 \pm 0.2 | 28.7 \pm 7.1 | 307.5 \pm 156.6 | 27.9 \pm 0.5 | 7.8 \pm 0.9 | 246.0 \pm 42.0 |
| Voting | 435 | 16 | 2 | 70.5 \pm 5.1 | 136.2 \pm 9.0 | 855.0 \pm 322.1 | 81.4 \pm 0.3 | 10.8 \pm 0.6 | 214.0 \pm 14.5 |
| Breast Cancer W. | 683 | 9 | 2 | 66.0 \pm 0.0 | 65.8 \pm 11.1 | 119.5 \pm 26.5 | 59.9 \pm 1.4 | 8.8 \pm 0.8 | 172.0 \pm 60.9 |
| Australian Credit | 690 | 14 | 2 | 51.2 \pm 0.1 | 54.3 \pm 7.2 | 138.5 \pm 28.9 | 49.9 \pm 0.4 | 6.6 \pm 0.3 | 89.5 \pm 21.3 |
| P.I.Diabetes | 768 | 8 | 2 | 17.8 \pm 0.0 | 68.9 \pm 13.3 | 120.0 \pm 21.9 | 17.6 \pm 0.1 | 6.9 \pm 0.3 | 100.5 \pm 19.1 |
| Concentric Rings | 800 | 4 | 4 | 15.7 \pm 0.2 | 49.2 \pm 11.7 | 101.0 \pm 62.8 | 15.2 \pm 0.5 | 5.4 \pm 1.1 | 75.0 \pm 52.2 |
| Parity 10-bits | 1024 | 10 | 2 | 26.6 \pm 0.0 | 32.1 \pm 5.6 | 22.5 \pm 6.8 | 26.6 \pm 0.0 | 17.7 \pm 4.8 | 209.0 \pm 243.7 |
| Average | | | | 39.3 | 33.3 | 189.3 | 37.6 | 6.9 | 133.4 |
| Wilcoxon p -value | | | | | | | 0.0106 | 0.0061 | 0.0879 |
| Large data | | | | | | | | | |
| Abalone | 4177 | 7 | 28 | 18.9 \pm 0.1 | 3148.4 \pm 609.8 | 184.0 \pm 59.6 | 15.2 \pm 0.2 | 29.8 \pm 1.3 | 73.0 \pm 13.4 |
| Spam | 4601 | 57 | 2 | 26.2 \pm 0.0 | 5260.5 \pm 105.6 | 105.5 \pm 2.8 | 25.3 \pm 0.2 | 184.7 \pm 4.1 | 102.0 \pm 3.5 |

by projection of data vectors on the first two directions $w_1^T x$ and $w_2^T x$ found by optimization of QPC1 and QPC2. The second direction w_2 have been found in the direction orthogonal to the first one. For the Australian dataset distinct separation between two groups of vectors is obtained. First projection on w_1 is sufficient to distinguish this two clusters. The Monk's 1 problem projected on the two dimensional space generated by QPC2 revealed inherent relations for this artificial dataset with symbolic features, leading to almost complete separation of instances with opposite labels. For the 10-bit parity problem both approaches found correct projections on diagonals of the hypercube representing Boolean function. In case of Concentric Rings noise has been suppressed and the two-dimensional ring structure hidden in this data was recovered.

3.2 Comparison of Generalization

The QPC projection index may be used for generation of new features that should reveal interesting aspects of analyzed data. Such features may be beneficial for training of almost any learning machines. Tab. 2 presents results obtained by training the Naive Bayes (NB) classifier with kernel density estimation on problems used for performance testing. First column contains results of NB trained on the original data. Each successive column represent results for NB trained on data projected on 1, 2 and 3 directions generated by the full (QPC1) index maximization and by its fast approximation (QPC2). Classification accuracy has been estimated using 10 fold stratified cross-validation repeated 10 times for each dataset and each method. To compare generalization of NB classifier trained with and without initial QPC transformation for each dataset corrected resampled t-test was used [9] and significant differences (at significance level 0.05) are marked with dots (see Tab. 2).

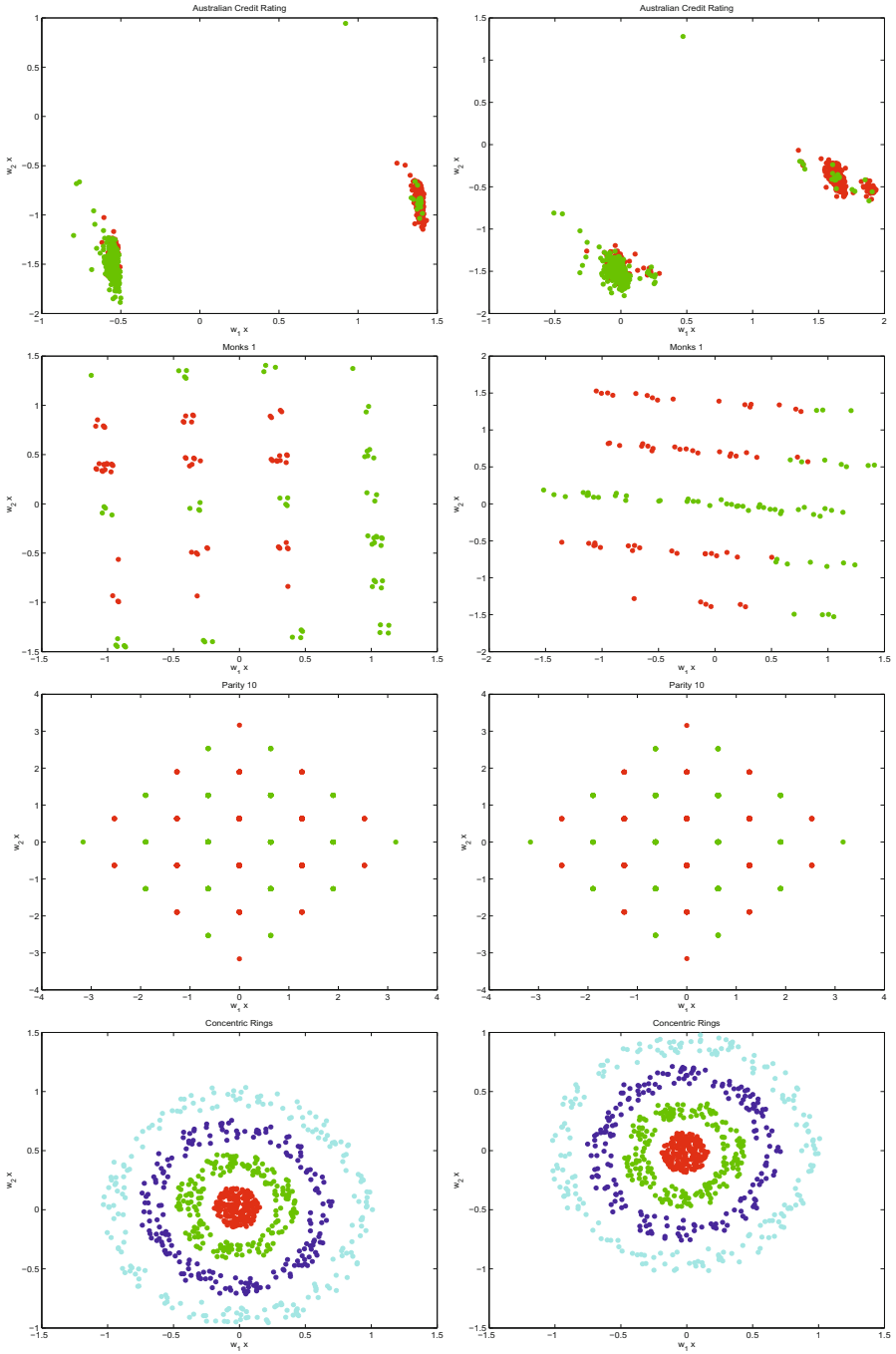


Fig. 1. Examples of the first two projections found by maximization of the full QPC1 index (left) and the approximated QPC2 index (right) for the Australian credit, the Monk's 1 problem, the 10-bit Parity and the Concentric Rings

Table 2. Average accuracy of the Naive Bayes with kernel density estimation in the 10x10 stratified CV test for the whole dataset and after training on dataset reduced to 1, 2 and 3 dimensions using two QPC versions

| Data set | Naive Bayes | QPC1+NB | | | QPC2+NB | | |
|---|-----------------|-------------------|-------------------|------------------|-------------------|-------------------|-------------------|
| | | 1 | 2 | 3 | 1 | 2 | 3 |
| Appendicitis | 84.4 \pm 10.2 | 87.4 \pm 8.2 | 86.1 \pm 8.8 | 84.9 \pm 9.6 | 87.1 \pm 8.9 | 86.0 \pm 9.2 | 86.1 \pm 9.0 |
| Monk's 1 | 71.5 \pm 11.3 | 71.3 \pm 11.0 | 82.7 \pm 13.9 ● | 89.2 \pm 9.4 ● | 67.2 \pm 12.7 | 82.9 \pm 13.0 ● | 87.9 \pm 11.0 ● |
| Iris | 95.7 \pm 4.9 | 98.0 \pm 4.0 | 95.9 \pm 5.2 | 95.8 \pm 5.2 | 96.9 \pm 4.6 | 95.9 \pm 5.2 | 96.0 \pm 5.1 |
| Wine | 97.7 \pm 3.5 | 92.5 \pm 5.8 ○ | 96.2 \pm 5.2 | 97.7 \pm 3.7 | 91.6 \pm 6.1 ○ | 97.4 \pm 4.0 | 97.6 \pm 3.8 |
| Ionosphere | 84.4 \pm 7.9 | 79.9 \pm 9.1 | 84.0 \pm 7.8 | 85.4 \pm 7.3 | 81.7 \pm 9.1 | 83.2 \pm 8.0 | 85.5 \pm 7.5 |
| Sonar | 75.8 \pm 10.1 | 74.1 \pm 10.4 | 75.4 \pm 10.1 | 75.8 \pm 9.3 | 73.3 \pm 10.5 | 75.9 \pm 10.4 | 76.5 \pm 9.0 |
| Glass | 60.3 \pm 9.9 | 55.3 \pm 8.3 | 56.0 \pm 8.7 | 59.9 \pm 8.9 | 54.8 \pm 9.8 | 56.5 \pm 9.7 | 59.1 \pm 9.8 |
| Heart Statlog | 79.8 \pm 7.3 | 80.2 \pm 7.2 | 82.8 \pm 6.8 | 82.6 \pm 7.2 | 80.5 \pm 7.5 | 82.7 \pm 7.0 | 83.0 \pm 7.1 |
| L.Breast | 72.7 \pm 6.1 | 72.3 \pm 5.3 | 72.6 \pm 6.6 | 73.7 \pm 6.4 | 70.6 \pm 6.3 | 70.8 \pm 7.2 | 70.6 \pm 8.0 |
| Heart Cleveland | 79.3 \pm 7.3 | 80.7 \pm 7.7 | 82.8 \pm 6.9 | 82.7 \pm 7.4 | 80.5 \pm 7.1 | 83.1 \pm 7.6 | 83.5 \pm 7.2 ● |
| Voting | 89.8 \pm 4.7 | 95.4 \pm 2.9 ● | 95.1 \pm 3.1 ● | 94.7 \pm 3.4 ● | 95.3 \pm 3.0 ● | 94.7 \pm 3.1 ● | 94.4 \pm 3.2 ● |
| Breast Cancer W. | 96.7 \pm 2.0 | 96.1 \pm 2.1 | 97.0 \pm 1.9 | 97.0 \pm 1.9 | 95.7 \pm 2.3 | 96.9 \pm 1.9 | 97.2 \pm 1.8 |
| Australian Credit | 68.4 \pm 6.0 | 85.3 \pm 4.7 ● | 85.5 \pm 4.4 ● | 86.2 \pm 4.7 ● | 85.4 \pm 4.5 ● | 85.4 \pm 4.4 ● | 85.8 \pm 4.4 ● |
| P.I.Diabetes | 73.6 \pm 5.1 | 76.4 \pm 4.4 ● | 74.9 \pm 4.5 | 73.9 \pm 5.2 | 76.3 \pm 4.5 | 73.9 \pm 4.6 | 72.7 \pm 5.1 |
| Concentric Rings | 85.9 \pm 3.6 | 64.0 \pm 4.3 ○ | 86.4 \pm 3.8 | 86.7 \pm 3.6 | 63.3 \pm 4.4 ○ | 84.9 \pm 4.9 | 85.6 \pm 4.0 |
| Parity 10 bits | 44.4 \pm 6.9 | 85.5 \pm 10.3 ● | 90.2 \pm 8.9 ● | 90.9 \pm 7.7 ● | 89.3 \pm 11.2 ● | 93.3 \pm 7.7 ● | 94.9 \pm 6.6 ● |
| Average | 78.8 | 80.9 | 84.0 | 84.8 | 80.6 | 84.0 | 84.8 |
| Win/Tie/Lose | | 4/10/2 | 4/12/0 | 4/12/0 | 3/11/2 | 4/12/0 | 5/11/0 |
| Wilcoxon NB vs. QPC+NB p -value | | 0.756 | 0.049 | 0.002 | 0.918 | 0.109 | 0.039 |
| Wilcoxon QPC1+NB vs. QPC2+NB p -value | | | | | 0.121 | 0.776 | 0.717 |

● - statistically significant improvement, ○ - statistically significant degradation

Features produced by QPC2 lead to similar accuracy to that of full QPC1. The Wilcoxon's signed-rank test shows no significant difference in accuracy of NB trained on the first three directions obtained by both QPC optimizations, giving p -value greater than 0.1 in all three cases (Tab. 2 last row). For all datasets t-test also shows no significant differences in NB accuracy between QPC1 and QPC2 transformation. In most cases NB trained on data projected on the first QPC direction produce results that are not significantly different from NB trained on the original data (10 ties obtained by corrected resampled t-test with level of significance equal to 5%). For 2 datasets t-test shows difference in accuracy in favor of original NB, but for 4 datasets the QPC transformations have improved NB generalization. For NB trained on data projected to the first two directions no significant degradation of accuracy is noted with comparison to NB trained on the original dataset. The Wilcoxon's signed-rank test confirms that there is no significant difference between accuracy of NB trained on first QPC projection and NB trained on original data, and there is significant difference in favor of NB trained on data projected to 2 or 3 dimensions obtained from QPC index both for QPC1 and QPC2. Thus a great reduction in dimensionality is obtained by using QPC features.

4 Discussion

The approximate version of the Quality of Projected Clusters projection pursuit method introduced in this paper greatly improve performance without degradation of the quality of results. As has already been stressed [10] separability is not the best goal of learning when problems are difficult, some intermediate tasks should be defined to derive

information that may help in finding optimal solutions. Many methods fail on difficult problems, such as the parity problem or the noisy concentric rings problem, but searching for good linear projection direction followed by simple one-dimensional nonlinear functions to distinguish pure clusters after the projection handles such problems without much effort. Therefore we are confident that such methods provide important computational intelligence tools.

Projections found by QPC may be used to enhance data representation expanding feature spaces (this was done in [11], where remarks on relations with kernel methods may be found). Each projection may also be implemented as a node in a hidden layer of feedforward network. This may be either followed by a simple linear layer (as in the multilayer perceptrons), or used only for initialization of weights. The prototypes obtained from QPC2 training may be directly used for classification as the nearest prototype vectors, or used for initialization in any radial-basis function method. The full QPC index has already been successfully applied to several constructive neural network architectures including QPC-NN [6] and QPC-LVQ [5]. The QPC-NN method build neural network optimizing QPC index within general sequential constructive method scheme proposed by [12]. The QPC-LVQ combines learning vector quantization [13] to map local relations with linear projections given by QPC to handle non local relations. Modification introduced in previous section should considerably increase performance of the QPC-based networks without loss of their generalization powers. Results of all these procedures will be presented in a longer paper in the near future.

Acknowledgment. This work was supported by the Polish Ministry of Higher Education under research grant no. N N516 500539.

References

1. Friedman, J.H., Tukey, J.W.: A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput.* 23(9), 881–890 (1974)
2. Friedman, J.: Exploratory projection pursuit. *Journal of the American Statistical Association* 82, 249–266 (1987)
3. Duch, W.: K-separability. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) *ICANN 2006*. LNCS, vol. 4131, pp. 188–197. Springer, Heidelberg (2006)
4. Grochowski, M., Duch, W.: Projection Pursuit Constructive Neural Networks Based on Quality of Projected Clusters. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) *ICANN 2008*, Part II. LNCS, vol. 5164, pp. 754–762. Springer, Heidelberg (2008)
5. Grochowski, M., Duch, W.: Constrained learning vector quantization or relaxed k-separability. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) *ICANN 2009*. LNCS, vol. 5768, pp. 151–160. Springer, Heidelberg (2009)
6. Grochowski, M., Duch, W.: Constructive Neural Network Algorithms that Solve Highly Non-Separable Problems. *Studies in Computational Intelligence*, vol. 258, pp. 49–70. Springer, Heidelberg (2010)
7. Asuncion, A., Newman, D.: UCI machine learning repository (2007), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
8. Wilcoxon, F.: Individual comparisons by ranking methods. *Biometrics* 1, 80–83 (1945)
9. Nadeau, C., Bengio, Y.: Inference for the generalization error. *Machine Learning* 52(3), 239–281 (2003)

10. Duch, W.: Towards comprehensive foundations of computational intelligence. In: Duch, W., Mandziuk, J. (eds.) *Challenges for Computational Intelligence*, vol. 63, pp. 261–316. Springer, Heidelberg (2007)
11. Maszczyk, T., Duch, W.: Support feature machines: Support vectors are not enough. In: *World Congress on Computational Intelligence*, pp. 3852–3859. IEEE Press, Los Alamitos (2010)
12. Muselli, M.: Sequential constructive techniques. In: Leondes, C. (ed.) *Optimization Techniques. Neural Network Systems, Techniques and Applications*, vol. 2, pp. 81–144. Academic Press, San Diego (1998)
13. Kohonen, T.: *Self-organizing maps*. Springer, Heidelberg (1995)