Basis exchange and learning algorithms for extracting collinear patterns

Leon Bobrowski\textsuperscript{a,b} and Paweł Zabielski\textsuperscript{a}

\textsuperscript{a}Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland; \textsuperscript{b}Institute of Biocybernetics and Biomedical Engineering, PAS, Warsaw, Poland

ABSTRACT
Understanding large data sets is one of the most important and challenging problems in the modern days. Exploration of genetic data sets composed of high dimensional feature vectors can be treated as a leading example in this context. A better understanding of large, multivariate data sets can be achieved through exploration and extraction of their structure. Collinear patterns can be an important part of a given data set structure. Collinear (flat) pattern exists in a given set of feature vectors when many of these vectors are located on (or near) some plane in the feature space. Discovered flat patterns can reflect various types of interaction in an explored data set. The presented paper compares basis exchange algorithms with learning algorithms in the task of flat patterns extraction.

ARTICLE HISTORY
Received 8 August 2018
Accepted 16 December 2020

KEYWORDS
Data exploration; flat patterns extraction; CPL criterion functions; basis exchange algorithms; learning algorithms

1. Introduction

Large data sets might contain very useful information. Models and computational tools for new knowledge extraction from large data sets are constantly being developed within pattern recognition or data mining methods (Duda et al., 2001; Hand et al., 2001). The contemporary problem of understanding large data sets is crucial for many practical issues. Better understanding of genetic data sets composed of high dimensional feature vectors is a leading example in this context (Wong, 1973).

We assume that explored data sets are composed of feature vectors which represent particular objects (events, phenomena) in a standardized way in a particular feature space. Data mining tools are used for discovering useful patterns in large data sets (Bobrowski, 2005). The term pattern denotes a subset (cluster) of feature vectors, characterized by a certain type of regularity (Duda et al., 2001). More specifically, the term collinear (flat) pattern means a large number of feature vectors situated on some plane in feature space.

Flat patterns can be extracted from large data sets through minimization of a special type of the convex and piecewise linear (CPL) criterion function (Bobrowski, 2005).
Basis exchange algorithms allow to find efficiently the global minimum of the CPL criterion function even in case of large, multidimensional large data sets. Learning algorithms provide another possibility of minimizing CPL functions (Bobrowski & Zabielski, 2016).

Comparison of a selected learning algorithm with a basis exchange algorithm in the task of flat patterns discovering is demonstrated in the presented paper. The results of the experimental comparison of these two types of algorithms are discussed.

2. Dual hyperplanes and vertices in the parameter space

Let us consider the data set \( C \) composed of \( m \) feature vectors \( x_j = [x_{j,1}, \ldots, x_{j,n}]^T \):

\[ C = \{x_j\}, \text{ where } j = 1, \ldots, m \] (1)

Feature vectors \( x_j \) belong to the \( n \)-dimensional feature space \( F[n] \) (\( x_j \in F[n] \)). Components \( x_{j,i} \) of the \( j \)-th feature vector \( x_j \) can be treated as numerical results of \( n \) standardized examinations of the \( j \)-th object \( O_j \), where \( x_{j,i} \in \mathbb{R} \) or \( x_{j,i} \in \{0,1\} \).

Each of \( m \) feature vectors \( x_j \) from the set \( C \) defines the following dual hyperplane \( h_j \) in the parameter space \( \mathbb{R}^n \) (Bobrowski, 2014):

\[ (\forall x_j \in C) \quad h^*_j = \{w : x_j^T w = 1\} \] (2)

where \( w = [w_1, \ldots, w_n]^T \) is the parameter (weight) vector (\( w \in \mathbb{R}^n \)).

Unit vectors \( e_i = [0, \ldots, 1, \ldots, 0]^T \) define the below hyperplanes \( h^0_i \) in the \( n \)-dimensional parameter space \( \mathbb{R}^n \) (Bobrowski, 2005):

\[ (\forall i \in \{1, \ldots, n\}) \quad h^0_i = \{w : e_i^T w = 0\} = \{w : w_i = 0\} \] (3)

Consider the set \( S_k \) of \( n \) linearly independent feature vectors \( x_j \) (1) and unit vectors \( e_i \):

\[ S_k = \{x_j \in J_k \cup^i : i \in I_k\} \] (4)

The set \( S_k \) contains \( r_k \) feature vectors \( x_j \) (\( j \in J_k \)) and \( n - r_k \) unit vectors \( e_i \) (\( i \in I_k \)).

The \( k \)-th vertex \( w_k = [w_{k,1}, \ldots, w_{k,n}]^T \) of the rank \( r_k \) in the parameter space \( \mathbb{R}^n \) is defined on the basis of the set \( S_k \) (4) as the intersection point of the \( r_k \) hyperplanes \( h_j \) (2) defined by the feature vectors \( x_j \) (\( j \in J_k \)) and the \( n - r_k \) hyperplanes \( h^0_i \) (3) defined by the unit vectors \( e_i \) (\( i \in I_k \)). The vertex \( w_k \) can be defined by the following linear equations:

\[ (\forall j \in J_k) \quad w_k^T x_j = 1 \] (5)

and

\[ (\forall i \in I_k) \quad w_k^T e_i = 0 \] (6)

The equations (5) and (6) can be represented as the matrix equation:

\[ B_k w_k = 1' = [1, \ldots, 1, 0, \ldots, 0]^T \] (7)

where the square, nonsingular matrix \( B_k \) is the \( k \)-th basis linked to the vertex \( w_k \):

\[ B_k = 1' = [x_{(1)0}, \ldots, x_{(r)k}, e_{(k+1)}, \ldots, e_{(n)}]^T \] (8)
The feature vector \( x_{j(l)} \) \((j(l) \in I_k)\) constitutes the \( l \)-th row of the matrix \( B_k \).

The vertex \( w_k \) can be computed in the below manner (7):

\[
w_k = B_k^{-1}1'
\]

The below three definitions are taken from the work (Bobrowski, 2014):

**Def. 2.1:** The rank \( r_k \) \((1 \leq r_k \leq n)\) of the \( k \)-th vertex \( w_k = [w_{k,1}, ..., w_{k,n}]^T \) (9) is defined as the number of the non-zero components \( w_{k,i} \) \( (w_{k,i} \neq 0) \).

**Def. 2.2:** The degree of degeneration \( d_k \) of the vertex \( w_k \) (9) of the rank \( r_k \) is defined as the number \( d_k = m_k - r_k \), where \( m_k \) is the number of such feature vectors \( x_j \) from the set \( C \) (1), which define the hyperplanes \( h_j \) (2) passing through this vertex \( (w_k^T x_j = 1) \). The vertex \( w_k \) (9) is degenerated if the degree of degeneration \( d_k \) is greater than zero \( (d_k > 0) \).

**Def. 2.3:** The basis \( B_k \) (8) is regular if and only if the number \( n - r_k \) of the unit vectors \( e_i \) \((i \in I_k) \) in the matrix \( B_k \) is equal to the number of such components \( w_{k,i} \) of the vector \( w_k = [w_{k,1}, ..., w_{k,n}]^T \) (9) which are equal to the zero \( (w_{k,i} = 0) \).

### 3. Vertexical planes and collinear patterns in feature space

The hyperplane \( H(w, \theta) \) in the feature space \( F[n] \) is defined as follows (Bobrowski, 2005):

\[
H(w, \theta) = \{x: w^T x = \theta\}
\]

where \( w \) is the weight vector \( (w \in \mathbb{R}^n) \) and \( \theta \) is the threshold \( (\theta \in \mathbb{R}^1) \).

The \( k \)-th vertex \( w_k \) (9) of the rank \( r_k \) \((r_k > 1) \) (Def. 1) allows to define the \((r_k - 1)\) - dimensional vertexical plane \( P_k(x_{j(1)}, ..., x_{j(r_k)}) \) in the feature space \( F[n] \) as the linear combination of \( r_k \) linearly independent feature vectors \( x_{j(i)} \) \((j(i) \in I_k) \) (5) from the set \( S_k \) (4) (Bobrowski, 2014):

\[
P_k(x_{j(1)}, ..., x_{j(r_k)}) = \{x \in F[n]: x = \alpha_1 x_{j(1)} + ... + \alpha_r x_{j(r_k)}\}
\]

where \( j(i) \in I_k \) (5) and the parameters \( \alpha_i \) \((\alpha_i \in \mathbb{R}) \) satisfy the following condition:

\[
\alpha_1 + ... + \alpha_r = 1
\]

Two linearly independent vectors \( x_{j(1)} \) and \( x_{j(2)} \) from the set \( C \) (1) support the below straight line \( l(x_{j(1)}, x_{j(2)}) \) in the feature space \( F[n] \) \((x \in F[n])\):

\[
l(x_{j(1)}, x_{j(2)}) = \{x: x = x_{j(1)} + \alpha(x_{j(2)} - x_{j(1)})\} = \{x: x = (1 - \alpha)x_{j(1)} + \alpha x_{j(2)}\}
\]

where \( \alpha \in \mathbb{R}^1 \).

The straight line \( l(x_{j(1)}, x_{j(2)}) \) (13) can be treated as the vertexical plane \( P_k(x_{j(1)}, x_{j(2)}) \) (11) spanned by two supporting vectors \( x_{j(1)} \) and \( x_{j(2)} \) with \( \alpha_1 = 1 - \alpha \) and \( \alpha_2 = \alpha \). In this case, the regular basis \( B_k \) (8) contains only two basis feature vectors \( x_{j(1)} \) and \( x_{j(2)} \) \((r_k = 2) \) and \( n - 2 \)
unit vectors \( \mathbf{e}_i \) (\( i \in \mathcal{I} \)). As a result, the vertex \( \mathbf{w}_k = [w_{k,1}, \ldots, w_{k,n}]^T \) (9) contains only two nonzero components \( w_{k,i} \) (\( w_{k,i} \neq 0 \)).

**Lemma 3.1:** The vertexical plane \( P_k(x_j(1), \ldots, x_{j(r_k)}) \) (11) based on the vertex \( \mathbf{w}_k \) (9) with the rank \( r_k \) greater than 1 (\( r_k > 1 \)) is equal to the hyperplane \( H(\mathbf{w}_k, 1) \) (10) in the \( n \)– dimensional feature space \( F[n] \) defined by this vertex.

**Theorem 3.1:** The \( j \)-th feature vector \( \mathbf{x}_j \) (\( \mathbf{x}_j \in \mathbb{C} \) (1)) is located on the vertexical plane \( P_k(x_j(1), \ldots, x_{j(r_k)}) \) (11) if and only if the \( j \)-th dual hyperplane \( h_j^1 \) (2) passes through the vertex \( \mathbf{w}_k \) (9) of the rank \( r_k \).

The proof of the above lemma and theorem was provided in the article (Bobrowski, 2014).

**Definition 3.4:** The collinear (flat) pattern \( F_k \) of the dimension \( r_k - 1 \) in the feature space \( F[n] \) is formed by \( m_k \) (\( m_k > r_k \)) feature vectors \( \mathbf{x}_j \) located on the vertexical plane \( P_k(x_j(1), \ldots, x_{j(r_k)}) \) (11). Lemma 1 shows that the flat pattern \( F_k \) is based on the vertex \( \mathbf{w}_k \) (9):

\[
F_k = \{ \mathbf{x}_j : \mathbf{w}_k^T \mathbf{x}_j = 1 \} \tag{14}
\]

The flat pattern \( F_k \) of the dimension \( r_k - 1 \) is composed of such \( m_k \) feature vectors \( \mathbf{x}_j \) that the dual hyperplanes \( h_j^1 \) (2) pass through the vertex \( \mathbf{w}_k \) (9) of the rank \( r_k \). The \( k \)-th vertex \( \mathbf{w}_k = [w_{k,1}, \ldots, w_{k,n}]^T \) (9) has \( n - r_k \) components \( w_{k,i} \) equal to zero:

\[
(\forall i \in \mathcal{I}_k(4)) \ w_{ki} = 0 \tag{15}
\]

4. Convex and piecewise linear (CPL) collinearity functions

Let us define the collinearity penalty functions \( \varphi_j^\varepsilon(\mathbf{w}) \) linked to feature vectors \( \mathbf{x}_j \) (1):

\[
(\forall \mathbf{x}_j \in \mathbb{C} \) (1)) \[
\varphi_j^\varepsilon(\mathbf{w}) = \begin{cases} 
1 - \varepsilon - x_j^T \mathbf{w} & \text{if } x_j^T \mathbf{w} < 1 - \varepsilon \\
0 & \text{if } 1 - \varepsilon \leq x_j^T \mathbf{w} \leq 1 + \varepsilon \\
x_j^T \mathbf{w} - 1 + \varepsilon & \text{if } x_j^T \mathbf{w} > 1 - \varepsilon 
\end{cases} \tag{16}
\]

where \( \varepsilon \geq 0 \) is the margin (\( \varepsilon \in \mathbb{R}^1 \)) (Figure 1).

**Remark 4.1:**

![Figure 1. The penalty functions \( \varphi_j^\varepsilon(\mathbf{w}) \) (16) with the margin \( \varepsilon \).](image)
The collinearity penalty functions \( \phi_j^\epsilon(w) \) (16) is equal to zero if and only if the distance between point \( w \) and the dual hyperplane \( h_j \) (2) is not greater than \( \frac{\epsilon}{|w_j|} \):

\[
\forall x_j \in C(1)) \text{ if } |x_j^T w - 1| \leq \epsilon, \text{ then } \phi_j^\epsilon(w) = 0 \quad (17)
\]

The \( k \)-th collinearity criterion function \( \Phi_k^\epsilon(w) \) is defined here as the sum of the penalty functions \( \phi_j^\epsilon(w) \) (16) linked to feature vectors \( x_j \) from the subset \( C_k \) (\( C_k \subset C(1) \)):

\[
\Phi_k^\epsilon(w) = \sum_{i \in J_k} \beta_j \phi_j^\epsilon(w) \quad (18)
\]

where \( J_k = \{ j : x_j \in C_k \subset C(1) \} \) and the positive parameters \( \beta_j (\beta_j > 0) \) can be treated as the prices of particular feature vectors \( x_j \). The standard choice of the parameters \( \beta_j \) values is one \( \{ (\forall j \in \{1, \ldots, m\}) \beta_j = 1.0 \} \).

5. The minimal value of the collinearity criterion function

The collinearity criterion function \( \Phi_k^\epsilon(w) \) (18) like the penalty functions \( \phi_j^\epsilon(w) \) (16) are convex and piecewise linear (CPL) (Bobrowski, 2014). It has been proved that the minimal value \( \Phi_k^\epsilon(w_\ast) \) of the convex and piecewise linear (CPL) criterion function \( \Phi_k^\epsilon(w) \) (18) can be found in one of the vertices \( w_k \) (9) (Simonnard, 1966):

\[
(\exists w_k^\ast)(\forall w) \Phi_k^\epsilon(w) \geq \Phi_k^\epsilon(w_k^\ast) \geq 0 \quad (19)
\]

Let us introduce the below layer \( L^\epsilon(w, \theta) \) in the feature space \( F[n] \) which is based on the hyperplane \( H(w, \theta) \) (10):

\[
L^\epsilon(w, \theta) = \{ \theta - \epsilon \leq w^T x \leq 1 + \theta \} \quad (20)
\]

where \( \epsilon \geq 0 \) is the margin \( (\epsilon \in \mathbb{R}^1) \).

The layer \( L^\epsilon(w, \theta) \) (20) contains such feature vectors \( x \) for which the distance from the hyperplane \( H(w, \theta) \) (10) is no greater than \( \epsilon / |w| \).

**Theorem 5.2:** If all the feature vectors \( x_j \) from the subset \( C_k \) (\( C_k \subset C(1) \)) are located inside the layer \( L^\epsilon(w, \theta) \) (20) with \( \theta \neq 0 \), then the minimal value \( \Phi_k^\epsilon(w_\ast) \) (19) of the collinearity criterion functions \( \Phi_k^\epsilon(w) \) (18) is equal to zero for the margin \( \epsilon' \) (16) no less than \( \epsilon / |\theta| \) \( (\epsilon' \geq \epsilon / |\theta|) \).

**Proof:**

If the vector \( x_j \) from the subset \( C_k \) is located inside the layer \( L^\epsilon(w, \theta) \) (20), then

\[
\theta - \epsilon \leq w^T x_j \leq 1 + \theta \quad (21)
\]

Let us suppose that \( \theta > 0 \). Then

\[
1 - \frac{\epsilon}{\theta} \leq \left( \frac{w}{\theta} \right)^T x_j \leq 1 + \frac{\epsilon}{\theta} \quad (22)
\]

In accordance with the definition (16) of the collinearity penalty function \( \phi_j^\epsilon(w) \):

\[
\phi_j^\epsilon \left( \frac{w}{\theta} \right) = 0 \quad (23)
\]
Because all the feature vectors $x_j$ from the subset $C_k$ are located inside the layer $L^e(w, 0)$ (20) with $\theta > 0$, then (18):

$$\varepsilon \Phi^e_k(\frac{w}{\theta}) = 0 \quad (24)$$

The last equation means that the minimal value of the collinearity criterion functions $\Phi^e_k(w)$ (18) with the margin $\varepsilon' = \varepsilon / |\theta|$ is equal to zero in the point $w^* = w / q$.

Let us suppose now that $\theta < 0$. By dividing both sides of the inequality (25) by the negative number $\theta$ we get the below result:

$$1 + \frac{\varepsilon}{|\theta|} \geq \frac{w}{|\theta|}^T x_j \geq 1 - \frac{\varepsilon}{|\theta|} \quad (25)$$

As a result, we obtain the below equation similar to (24):

$$\varepsilon \Phi^e_k(\frac{w}{|\theta|}) = 0 \quad (26)$$

The minimal value $\Phi^e_k(w_k^*)$ (19) of the collinearity criterion function $\Phi^e_k(w)$ (18) with $\varepsilon' = \varepsilon / |\theta|$ is equal to zero. An increase of the margin $\varepsilon$ in the penalty functions $\phi^e_j(w)$ (16) allows to preserve the location of all vectors $x_j$ in the layer $L^e(w, 0)$ (20). So, the minimal value $\Phi^e_k(w_k^*)$ (19) remains equal to zero.

It can be proved that the minimal value $\Phi^e_k(w_k^*)$ (19) of the CPL criterion function $\Phi^e_k(w)$ (18) can be found in one of the vertices $w_k$ (Simonnard, 1966). So, taking into account the equation (26) we infer that such vertex $w_k^*$ (9) exists for which $\Phi^e_k(\frac{w}{|\theta|}) = 0$.

6. Basis exchange algorithm

The basis exchange algorithm was initially proposed and developed as an efficient tool for designing linear classifiers and examining linear separability of large, multidimensional data sets (Bobrowski, 2005). The first version of the basis exchange algorithm was described in the papers (Bobrowski, 1991; Bobrowski & Niemiro, 1984). The first basis exchange algorithms were aimed at minimizing the perceptron criterion function. The convex and piecewise linear (CPL) perceptron criterion function links the linear separability concept, which is fundamental in the perceptron theory of neural networks (Duda et al., 2001). The CPL criterion functions were specified later for variety of goals. Different types of basis exchange algorithms were proposed for particular CPL criterion functions (Bobrowski, 2005).

The basis exchange algorithms are based on the Gauss-Jordan transformation. The famous Simplex algorithm used in the linear programming is also based on this transformation (Simonnard, 1966). A brief description of the basis exchange algorithms aimed at the minimization of the function $\Phi^e_k(w)$ (18) is presented below.

Let us assume that the $k$-th basis $B_k$ (8) is the squared matrix composed from $n$ linearly independent feature vectors $x_{j(i)}$ from the subset $C_k (C_k \subset C \ (1))$:

$$B_k = [x_{j(1)}, \ldots, x_{j(n)}]^T \quad (27)$$
The inverse matrix $B_k^{-1}$ during the $k$-th stage can be represented in the below manner:

$$B_k^{-1} = [r_{1(k)}, \ldots, r_{n(k)}]$$  \hspace{1cm} (28)

The vectors $x_j(i)$ and $r_i(k)$ fulfil the below equations:

$$\forall i, i' \{1, \ldots, n\} \quad x^T_j(i)r_i(k) = 1$$  \hspace{1cm} (29)

and

$$\text{if } i \neq i', \text{ then } x^T_j(i)r_i(k) = 0$$

During the $k$-th stage of the algorithm the basis $B_k$ (27) is changed into the basis $B_{k+1}$. The matrix $B_{k+1}$ is created from the matrix $B_k$ (30) through replacing of the $l$-th row $x_j(l)$ by the new vector $x_k$ taken from a given data subset $C_k \subset C$ (1)).

The exchange of the $l$-th basis vector $x_j(l)$ by the new vector $x_k$ results in the new basis $B_{k+1}$ (30) and the new inverse matrix $B_{k+1}^{-1} = [r_1(k+1), \ldots, r_n(k+1)]$ (28). The Gauss–Jordan transformation allows to compute efficiently the columns $r_i(k+1)$ of the matrix $B_{k+1}^{-1}$ on the basis of the columns $r_i(k)$ (1) of the inverse matrix $B_k^{-1}$ (28), where $k = 0, 1, \ldots, K$ (Bobrowski, 1991; Bobrowski & Niemiro, 1984):

$$r_{i(k+1)} = \left( \frac{1}{r^T_{i(k)}x_k} \right) r_{i(k)}$$  \hspace{1cm} (30)

and $\forall i \neq l$

$$r_{i(k+1)} = r_{i(k)} - (r^T_{i(k)}x_k)r_{i(k+1)} = r_{i(k)} - (r^T_{i(k)}x_k/r^T_{i(k)}x_k)r_{i(k)}$$

The basis exchange algorithms are based on the Gauss–Jordan transformation (30). The index $l$ of the vector $x_j(l)$ which leaves the basis $B_k$ (27) during the $k$-th stage is determined by the exit criterion of a particular basis exchange algorithm. The entry criterion determines which vector $x_k$ from the data subset $C_k$ (1) enters the new basis $B_{k+1}$ (27). The stop criterion allows to determine the final stage $K$ of the basis exchange algorithm. Selected criteria (exit, entry, and stop criterion) should ensure the collinearity function $\Phi_k^e(w)$ (21) decreasing during each stage $k$.

The exchange of the $k$-th basis $B_k$ (27) into the basis $B_{k+1}$ causes the move from the vertex $w_k$ to $w_{k+1}$ (9). After finite number $K$ of steps the optimal vertex $w_k^*$ (19) is reached this way (Bobrowski, 2005).

7. Learning algorithm

The optimal vertex $w_k^*$ (9) which constitutes the minimum $\Phi_k^e(w_k^*)$ (19) of the collinearity criterion function $\Phi_k^e(w)$ (18) can be approximated using learning algorithms (Kushner & Yin, 1997). The Robbins – Monro procedure of the stochastic approximation has been used by us in designing learning algorithm aimed at the criterion function $\Phi_k^e(w)$ (18) minimization (Bobrowski and Zabielski). For this purpose it is useful to represent the criterion function $\Phi_k^e(w)$ (18) as the regression function $\Phi_k(w)$ with the uniform probability distribution $p(x_j)$ (Tsypkin, 1973):

$$\Phi_k(w) = \sum_{j \in J_k} p(x_j)q_j^e(x_j; w)$$  \hspace{1cm} (31)
where $\phi^c(x_j; w) = \phi^c_j(w)$ (16), and $p(x_j)$ is the uniform probability distribution defined on $m_k$ elements $x_j$ of the $k$-th data subset $C_k$ ($C_k \subset C$ (1)) as follows:

$$p(x_j) = \frac{1}{m_k}$$

The uniform probability distribution $p(x_j)$ was used to generate the below learning sequence $\{x(l)\}$ from $m_k$ elements $x_j$ of the $k$-th data subset $C_k$:

$$\{x(1), x(2), \ldots, x(n), \ldots\}$$

The elements $x(l)$ of the above learning sequence were generated in each step $l$ independently in accordance with the probability distribution $p(x_j)$ (32):

$$(\forall l = 1, 2, 3, \ldots) \quad P\{x(l) = x_j\} = p(x_j) = \frac{1}{m_k}$$

where $P\{x(l) = x_j\}$ is the probability of obtaining the $j$-th vector $x_j$ during the $l$-th learning step.

The Robbins–Monro procedure aimed at minimization of the regression function $\Phi_k(w)$ (31) can be given in the below manner (Tsypkin, 1973):

$$w(l + 1) = w(l) - \alpha_l \nabla_w \Phi^c(x(l); w(l))$$

where $w(l)$ is the value of the weight vector during the $l$-th learning step, and $\{\alpha_l\}$ is a sequence of decreasing, positive parameters $\alpha_l$, e. g. $\alpha_l = 1/l$ which fulfil the below

Figure 2. Two–dimensional learning set $D_3$ with 66% outliers.
conditions:

\[ \sum_{l} \alpha_l = \infty \quad \text{and} \quad \sum_{l} \alpha_l^2 < \infty \]  

(36)

The Robbins – Monro procedure (35) applied to the regression function \( \Phi_k(w) \) (31) with the penalty functions \( \phi^\varepsilon(x_j; w) = \phi_j^\varepsilon(w) \) (16) allows to specify the below learning algorithm:

\[
\forall l = 1, 2, \ldots \\
\text{if } w(l) \mathbf{x}(l) \mathbf{T} < 1 - \varepsilon, \text{ then } w(l+1) = w(l) + \alpha_l \mathbf{x}(l), \text{ and} \\
\text{if } w(l) \mathbf{x}(l) \mathbf{T} > 1 + \varepsilon, \text{ then } w(l+1) = w(l) - \alpha_l \mathbf{x}(l), \text{ and} \\
\text{if } 1 - \varepsilon \leq w(l) \mathbf{x}(l) \mathbf{T} \leq 1 + \varepsilon, \text{ then } w(l+1) = w(l)
\]  

(37)

where \( \mathbf{x}(l) = \mathbf{x}_j \) if the feature vector \( \mathbf{x}_j \) was generated (34) during the \( l \)-th learning step.

---

**Figure 3** Distances \( \rho(w(n), w_k^*) \) between the result \( w(n) \) during the \( n \)-th step and the optimal solution \( w_k^* \) (22) on the two-dimensional learning sets \( D_1 \) (squares), \( D_2 \) (circles) and \( D_3 \) (triangles) with different level of outliers (33%, 50%, and 66% respectively). The experiments were carried out using an iterative learning algorithm (35).
The learning algorithm (37) can be treated as a modification of the famous error-correction algorithm used in the Perceptron (Duda et al., 2001)

**Theorem 7.3:** If the learning sequence \(\{x(l)\}\) (33) is generated in accordance with the stationary probability distribution \(p(x_j)\) (34), then sequence \(\{w(l)\}\) of the weight vectors \(w(l)\) generated with the learning algorithm (37) converges to the optimal vector \(w_k^*\), which constitutes the minimal value \(\Phi_k(w_k^*)\) (19) of the criterion function \(\Phi_k(w)\) (18).

The proof of this theorem can be given using theorems of the stochastic approximation method (Kushner & Yin, 1997; Tsypkin, 1973).

**8. Experimental results**

The compared algorithms of the basis exchange (30) and the iterative learning algorithm (35) were implemented in C++ and run on an Intel Core i7, 1.73 GHz processor (turbo boost to 2.93 GHz).

![Figure 4. Distance \(\rho(w(n), w_k^*)\) between the result \(w(n)\) during the \(n\)-th step and the optimal solution \(w_k^*\) (19) defined on the two-dimensional learning data sets \(D_1\) (squares), \(D_2\) (circles) and \(D_3\) (triangles) with different levels of outliers. Experiments were performed using the basis exchange algorithm (30).](image)
For the computational experiments, three synthetic datasets $D_1$, $D_2$ and $D_3$, were created, each with a different level of outliers. The term outliers here denotes such additional feature vectors $x_j$ which were not located on the vertexical line $l_k(x_j(1), x_j(2))$ (13). The outlier feature vectors $x_j$ were generated in the two-dimensional feature space $R^2$ according to the normal distribution $N_2(0, I)$ with the unit covariance matrix $I$. The levels of outliers were established as follows: $D_1$ – 33%, $D_2$ – 50% and $D_3$ – 66%. One of these datasets is shown in Figure 2.

Figure 3 shows results of the learning process with the iterative learning algorithm as the distance $\rho(w(n), w_k^*)$ between the result $w(n)$ (35) during the $n$-th step and the optimal solution $w_k^*$ (22) defined on data sets $D_k$ with different level of outliers.

As shown in these figures, the iterative algorithm converges after approximately $n = 200$ steps. However, in case of higher level of outliers, the solution is worse and the distance between the result and the optimal solution $w_k^*$ (19) is greater. Algorithm ends after $n = 1000$ steps in case of each data set $D_k$.

For comparison, Figure 4 shows the results of the basis exchange algorithm, where the plot demonstrates the distance $\rho(w(n), w_k^*)$ between the result $w(n)$ (40) during the $n$-th step and the optimal solution $w_k^*$ (19).

As shown in these figures, a solution has been found in each case for data with different level of outliers. The distances $\rho(w(n), w_k^*)$ between the result $w(n)$ during the $n$-th step and the optimal solution $w_k^*$ (19) became equal to zero.

The basis exchange algorithm performed worse in the first few steps than the iterative algorithm. But after a few steps and finally the basis exchange algorithm improved and

![Figure 5](image-url). The two-dimensional learning set $D_4$ with the margin $\delta = 0.1$
performed better results than iterative algorithm. Moreover, the iterative algorithm needs more time to find a solution than the basis exchange algorithm.

The next experiments concerned the role of the margin parameter $\varepsilon$ (16). The iterative learning algorithm (35) and the database exchange algorithm (30) with different values of the parameter $\varepsilon$ were tested on the data set shown in Figure 5.

The results of the learning process based on the iterative learning algorithm (35) are presented in Figure 6. The algorithm (35) was stopped when the condition: $1 - \varepsilon \leq w(n)^T x(n) \leq 1 + \varepsilon$ was reached.

These results show that the iterative learning algorithm converges after about 20 steps. The solution becomes worse in cases of a higher value of the examined parameter $\varepsilon$. If a margin greater than zero ($\varepsilon > 0$) is selected, the algorithm ends faster than in the case of algorithms without a margin ($\varepsilon = 0$).

Similarly, Figure 7 shows the results of the basis exchange algorithm (30) with different values of the margin $\varepsilon$. (16).

The final values of the criterion function $\Phi_k(w)$ (18) for the basis exchange algorithm are equal to or close to zero. It gives better results and works faster than iterative algorithm. The computation ends earlier for a margin $\varepsilon$ greater than zero.

Figure 6. Distance $\rho(w(n), w^*_k)$ between the result $w(n)$ during the $n$-th step and the optimal solution $w_k^*$ (19) defined on the two-dimensional learning data set $D_4$ with different margin $\varepsilon$ ($\varepsilon = 0$ represented by squares, $\varepsilon = 0.5\delta$ represented by circles, $\varepsilon = \delta$ represented by triangles. Experiments were carried out using the iterative learning algorithm (35).
The final values of the criterion function $\Phi_k^\varepsilon(w)$ (18) for the base exchange algorithm are equal to or close to zero. The basis exchange algorithm gives better results and runs faster than the iterative algorithm. The computation ends earlier for a margin $\varepsilon$ greater than zero.

Figures 8–10 show the dependence of the result on the value of the margin $\varepsilon$. As the value of the margin $\varepsilon$ increases, more feature vectors $x_j$ (1) are in the solution layer $L^\varepsilon(w, \theta)$ (20).

**7. Concluding remarks**

A better understanding of large, multivariate data sets can be achieved through identification of collinear (flat) patterns $F_k$ (14). Extraction of collinear patterns $F_k$ (14) can be performed through the minimization of the collinear criterion function $\Phi_k^\varepsilon(w)$ (18) defined on various data subsets $C_k$ ($C_k \subset C$ (1)).

Both the basis exchange algorithm (30) as well as the iterative learning algorithm (35) can be used in the minimization of the collinear criterion function $\Phi_k^\varepsilon(w)$ (18). Usability of both types of algorithms was tested experimentally. Selected results of these experiments are shown on Figures in the article.
Particular attention in the article was paid to the role of the margin $\varepsilon$ (16) in extraction of flat patterns $F_k$ (14). The margin $\varepsilon$ greater than zero ($\varepsilon > 0$) allows to discover not only collinear patterns $F_k$ (17) as the subsets of points $x_j$ located on vertexical planes $P_k(x_j^{(1)})$.

**Figure 8.** The result of the data set $D_4$ with value of parameter $\varepsilon = 0$.

**Figure 9.** The result do the data set $D_4$ with value of parameter $\varepsilon = 0.5*\delta$.  

Particular attention in the article was paid to the role of the margin $\varepsilon$ (16) in extraction of flat patterns $F_k$ (14). The margin $\varepsilon$ greater than zero ($\varepsilon > 0$) allows to discover not only collinear patterns $F_k$ (17) as the subsets of points $x_j$ located on vertexical planes $P_k(x_j^{(1)})$. 
… $x_{j(rk)}$ (11), but also flat patterns located in layers $L^\varepsilon(w, \theta)$ (23) of different thicknesses. Noise reduction may be based on such layered aggregation of data subsets $C_k$. Increasing capacity in this area should be achieved by applying the RLS method of feature subset selection (Bobrowski & Łukaszuk, 2011).

The proposed method of discovering collinear patterns on the basis of the CPL criterion functions $\Phi_k^\varepsilon(w)$ (18) can be treated as a generalization of the methods based on the Hough transformation. The Hough transformation techniques are used in computer vision for detection lines and curves in pictures (Ballard, 1981; Duda & Hart, 1972).

**Disclosure statement**

No potential conflict of interest was reported by the author(s).

**Notes on contributors**

Professor Leon Bobrowski works at the Computer Science Faculty of Bialystok University of Technology and, additionally, in the Laboratory of Biomedical Data Analysis at the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences in Warsaw. His research interests include data exploration, bioinformatics, and machine learning. He is developing data mining techniques based on the minimization of the convex and piecewise linear (CPL) criterion functions. The CPL criterion functions are linked to the concept of the linear separability of large,
multivariate data sets and to the perceptron model of neuronal networks. The relaxed linear separability (RLS) method of feature subset selection has been developed on the basis of the CPL functions and has been used for searching specific genes. The basis exchange algorithms have been developed and are used for efficient and precise minimization of the CPL criterion functions. These algorithms are based on the Gauss-Jordan transformation and, for this reason, are similar to the Simplex algorithm from linear programming.

Paweł Zabielski graduated at the Faculty of Computer Science of the Białystok University of Technology in Poland with an excellent result. He teaches the following subjects: Basics of programming, Object-oriented programming, Programming applications in JavaScript, Programming, algorithms and data structures, Advanced programming techniques, Building applications in WPF technology, Analysis and testing of information systems. Under the supervision of Professor Leon Bobrowski, he started research work related to the subject of detection of flat patterns. The subject of his research work is focused on the analysis of large data sets and the detection of certain patterns. He develop methods and algorithms for acquiring knowledge. The results of his work have been published in scientific articles.

References

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2), 111–122. https://doi.org/10.1016/0031-3203(81)90009-1

Bobrowski, L. (1991). Design of piecewise linear classifiers from formal neurons by some basis exchange technique. *Pattern Recognition*, 24(9), 863–870. https://doi.org/10.1016/0031-3203(91)90005-P

Bobrowski, L. (2005). Data mining based on convex and piecewise linear criterion functions (in Polish). Technical University of Białystok.

Bobrowski, L. (2014). Discovering main vertexical planes in a multivariate data space by using CPL functions. In P. Perner (Ed.), *ICDM 2014*. Springer Verlag.

Bobrowski, L., & Łukaszuk, T. (2011). Relaxed linear separability (RLS) approach to feature (gene) subset selection. In X. Xia (Ed.), *Selected works in bioinformatics* (pp. 103–118). INTECH.

Bobrowski, L., & Niemiro, W. (1984). A method of synthesis of linear discriminant function in the case of nonseparability. *Pattern Recognition*, 17(2), 205–210. https://doi.org/10.1016/0031-3203(84)90059-1

Bobrowski, L., & Zabielski, P. (2016). Learning Algorithms Aimed at Collinear Patterns in: *ACIIDS* (2) (pp. 159–168). Springer Verlag.

Duda, O. R., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of Association for Computing Machinery*, 15(1), 11–15. https://doi.org/10.1145/361237.361242

Duda, O. R., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. J. Wiley.

Hand, D., Smyth, P., & Mannila, H. (2001). *Principles of data mining*. MIT Press.

Kushner, H. J., & Yin, G. G. (1997). *Stochastic approximation algorithms and applications*. Springer Verlag.

Simonnard, M. (1966). *Linear programming*. Prentice-Hall.

Tsypkin, Y. Z. (1973). *Foundations of the theory of learning systems*. Academic Press.

Wong, K.-C. (ed.). (1973). *Big data analytics in genomics*. Academic Press. Springer Verlag, Berlin (2016).