Optimized Bi-Dimensional Data Projection For Clustering Visualization

Rodrigo T. Peres[a], Claus Aranha[c,*], Carlos E. Pedreira[a,b]

[a] *COPPE-PEE – Engineering Graduate Program, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.*
[b] *Faculty of Medicine, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.*
[c] *Graduate School of Systems and Information Engineering, The University of Tsukuba, Tsukuba, Japan.*

Abstract

We propose a new method to project n-dimensional data onto two dimensions, for visualization purposes. Our goal is to produce a bi-dimensional representation that better separate existing clusters. Accordingly, to generate this projection we apply Differential Evolution as a meta-heuristic to optimize a divergence measure of the projected data. This divergence measure is based on the Cauchy-Schwartz Divergence, extended for multiple classes. It accounts for the separability of the clusters in the projected space using the Renyi entropy and Information Theoretical Clustering analysis. We test the proposed method on two synthetic and five real world data sets, obtaining well separated projected clusters in two dimensions. These results were compared with results generated by PCA and a recent likelihood based visualization method.

Keywords: Visualization, Pattern Analysis, Information Theoretical Learning, Parameter Learning, Evolutionary Computation

1. Introduction

The problem of Data Visualization consists of generating a bi-dimensional projection of a high-dimensional data set. One of its aims is to make the different classes in the original data set be shown as distinct clusters in a bi-dimensional projection. In this sense, a good projection will have two desirable characteristics: no instances of a certain class in the original data set will be placed in the cluster of a different class in the projection, and the clusters for the different classes should be well defined and separated from each other.

An optimized (concerning classification) bi-dimensional projection of the data may allow a visual inspection of the data sets, retrieving information about

[*] Corresponding author

Email addresses: `rperes@lps.ufrj.br` (Rodrigo T. Peres), `caranha@cs.tsukuba.ac.jp` (Claus Aranha), `pedreira@ufrj.br` (Carlos E. Pedreira)

shape and separation of the clusters. When new data instances are presented to this calculated projection, it is possible to easily assess how strongly the new data points are associated to the classes by looking at their location in the projected space. Even in case a data point lays far away from all the clusters, this may lead the professional examining the data to suspect the existence of a new class not previously considered, or that this data point does not in fact belong to any of the groups. In other words, a good projection is a tool for increasing the understanding of a data set.

Data visualization is very important for medical applications. One such example is flow cytometry, a widely used technique essential to the diagnosis and follow-up of a wide spectrum of diseases, including HIV-infection and clonal hematological disorders such as acute and chronic leukemias and non-Hodgkin's lymphomas [16]. Flow cytometry data sets contain from tens of thousands to millions of observations, with dozens of attributes. Due to the unique way in which a disease presents itself in different patients, it is important not only to classify a particular data point as one of multiple sub-classes, but also to observe how far it deviates from the known clusters. For this task, observation of projected data sets by experienced practitioners is essential. There is a lot of interest in applying computational methods for the analysis of flow cytometry problems [3, 17, 18].

An early proposal of the use of projections to visualize multidimensional data was the Grand Tour [1]. The Grand Tour uses a series of rotations of an orthogonal bi-dimensional projection to provide multiple views of the same data. However, it does not include a criterion for choosing one projection over another. One such criterion is the PCA [11], where the two directions with greatest variance are used as the bi-dimensional projection. Supervised criteria have also been used to choose the best projection for visualization. We mention among these the LDA [8], and the LF [30].

A different family of visualization methods includes multidimensional scaling (MDS) [22] and its variations, such as the supervised MDS [27]. These methods are essentially different from the previous ones in that they calculate the bi-dimensional data points based on the distances among the observations, instead of making a linear transformation of the original attributes.

In this paper, we propose a new method for generating bi-dimensional projections for data visualization, aiming at better cluster separation on $\Re^2$. A Differential Evolution algorithm is used to generate projections with an optimal value for the Cauchy-Schwartz divergence measure.

Differential Evolution (DE) is a meta-heuristic for parameter optimization [19, 23]. Based on Evolutionary Algorithms, the DE creates a random set of candidate solutions to the optimization problem, and mixes the best performing solutions through a set of genetics-based operations (mutation and crossover).

Recently, DE has seen a lot of use in the fields of data clustering [6, 14] and classification [5]. The problem of Data Visualization is closely related to the clustering and classification problems, in that it is important to define a measure of divergence between data points, and determine whether data clusters contain points of a single label. Because of this, DE success in these field is of special interest to us. In particular, DE has been shown to be robust regarding noise in the classification domain [12], something that is also hinted at in the results of this work.

The proposed method is tested on a number of experiments, both from synthetic and real world data sets. The results of these experiments are compared with those of the classic PCA and a newer visualization method based on the optimization of a likelihood measure.

The results indicate that the proposed method is able to generate projections with good cluster separation. The proposed method is robust regarding initial conditions and noisy attributes.

2. Methodology

Let X be a sample of size n comprised of a set of m-dimensional observations $\{x_1, x_2, ..., x_n; x_i \in \Re^m, i = 1, ...n\}$. To each observation x_i we assign one out of k possible labels, $L_1, L_2, ...L_k$.

The problem of visualization as described in this paper consists of finding a function $P(x) = x', P : \Re^m \rightarrow \Re^2$, where m is the number of dimensions in the original data set. In other words, a "projection function" that transforms a point x in $\Re^m$ into a point x' in $\Re^2$.

A successful projection function P is one that, given a point $x \in \Re^m$ belonging to class L_i, then $P(x) \in \Re^2$ will be close to other points labeled as L_i and distant from points labeled as $L_j, i \neq j$. Such a projection is said to have well separated clusters corresponding to each label.

A key issue when addressing this problem is how to define "well separated clusters"[10]. Divergence measures allow the quantification of the difference between two probability distributions [4]. Let us consider a divergence measure $D(X')$ that increases as the clusters in X' are well separated. In this case, the visualization problem turns into finding the projection function $P(X)$ that maximizes $D(P(X))$.

To solve this problem we propose the use of a Differential Evolution algorithm to generate bi-dimensional projections that maximize the Cauchy-Schwartz divergence measure among the projected clusters. Each component will be detailed in the following subsections.

2.1. Cauchy-Schwartz Divergence Measure

We use the distance between probability distribution functions to quantify how far one cluster is from another. Accordingly, we propose the use of the Cauchy-Schwartz divergence D_{C-S} [28] as a measure of the distance between clusters in the projected space where X' lays. For two probability distribution functions (pdf's) p and q, D_{C-S} is calculated as

$$\begin{aligned} D_{C-S}(l_1, l_2) = & -\log \frac{(\int p(x)q(x)dx)^2}{\int p^2(x)dx \int q^2(x)dx} \\ = & \log \int p^2(x)dx + \log \int q^2(x)dx - 2\log \int p(x)q(x)dx. \end{aligned} \quad (1)$$

In this equation, $\log \int p^2(x)dx$ is the estimation for the quadratic density of p. This is equivalent to the negative of the Renyi quadratic entropy of p. In the same way, $-2\log \int p(x)q(x)dx$ measures the interaction between the two pdf's p and q.

Note that, clearly, $D_{C-S}(p,p) = 0$ and $D_{C-S}(p,q) = D_{C-S}(q,p)$ for any pdf's p and q. Besides, $D_{C-S}(p,q) \geq 0$ [21, 28]. Nevertheless, it is worth noting that the D_{C-S} is not a metric, since it does not satisfy the triangle inequality [28].

Let us consider a data set with two associated labels, L_1 and L_2, as having been generated by two pdf's p and q resulting in the projected observations X_1' and X_2', respectively. It follows that the distance between clusters associated to L_1 and L_2 may be measured through $D_{C-S}(p,q)$. Therefore, if there are two projections where the D_{C-S} value calculated for the second projection is greater than the one calculated for the first, then the clusters in the second projection are better separated than those in the first one.

The divergence calculation implies, in general, the necessity of estimating the involved pdf's. This can be especially hard for continuous problems, due to the need of some sort of discretization procedure. The Information Theoretic Learning (ITL) approach overcomes this setback. It was proposed in [20], and is briefly described below.

Let N_1 and N_2 be the sizes of the samples generated by pdf's p and q, respectively. These pdf's may be estimated through a Parzen Windows approach [7]. Accordingly, let $G_{\sigma^2}(x)$ be the Gaussian probability function with zero mean and variance σ^2

$$G_{\sigma^2} = \frac{1}{\sqrt{2\sigma^2\pi}}\exp(-\frac{x^2}{2\sigma^2}). \tag{2}$$

An estimate $\hat{p}$ of the pdf p, using $G_{\sigma_p^2}$ as kernel, is given by

$$\hat{p} = \frac{1}{N_1}\sum_{i=1}^{N_1} G_{\sigma_p^2}(x - x_i). \tag{3}$$

It follows that

$$\begin{aligned}
&\int \hat{p}(x)\hat{q}(x)dx = \\
&= \int \frac{1}{N_1}\sum_{i=1}^{N_1} G_{\sigma_p^2}(x - x_i)\frac{1}{N_2}\sum_{j=1}^{N_2} G_{\sigma_q^2}(x - x_j)dx \\
&= \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2}\int G_{\sigma_p^2}(x - x_i)G_{\sigma_q^2}(x - x_j)dx \\
&= \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} G_{\sigma_p^2+\sigma_q^2}(x_i - x_j).
\end{aligned} \tag{4}$$

The last equality in eq. 4 results from the convolution theorem for Gaussians [21, 25]. It is quite interesting to note that the final expression of eq. 4 depends exclusively on the observations x_i and the kernel width $(\sigma_p^2 + \sigma_q^2)$.

In a similar manner we have

$$\int \hat{p}^2(x)dx = \frac{1}{N_1^2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} G_{2\sigma_p^2}(x_i - x_j) \tag{5}$$

and

$$\int \hat{q}^2(x)dx = \frac{1}{N_2^2} \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} G_{2\sigma_q^2}(x_i - x_j). \tag{6}$$

By applying eqs. 4, 5, and 6 in eq. 1, the C-S divergence between pdf's p and q can be calculated as

$$\begin{aligned} D_{C-S}(\hat{p}, \hat{q}) = {} & \log \frac{1}{N_1^2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} G_{2\sigma_p^2}(x_i - x_j) \\ & + \log \frac{1}{N_2^2} \sum_{i=1}^{N_2} \sum_{j=1}^{N_2} G_{2\sigma_q^2}(x_i - x_j) \\ & - 2 \log \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} G_{\sigma_p^2 + \sigma_q^2}(x_i - x_j). \end{aligned} \tag{7}$$

The D_{C-S} was originally proposed for cases with two labels. We define a simple extension of the D_{C-S} measure for k pdf's $p_1, p_2, ...p_k$. Each pdf is associated to one of k, and the number of observations in X associated with each label is n_i $(\sum_i^k n_i = n)$.

Let us divide eq. 7 in two parts, the first where we calculate the estimation of the quadratic density for each pdf (composed of eqs. 5 and 6), and the second where we calculate the interaction between the pdf's (eq. 4).

We can extend the first part for k labels by adding the estimation of the quadratic density of the pdf's associated to every label. First, let's define x_{li} as the i-th element in X with label l. Now, for each label $l \in 1, 2, ..., k$, we calculate

$$H(l) = \log \frac{1}{n_l^2} \sum_{i=1}^{n_l} \sum_{j=1}^{n_l} G_{2\sigma_l^2}(x_{li} - x_{lj}). \tag{8}$$

The second part of eq. 7 calculates the Clustering Evaluation Function (CEF), as defined in [9]. The CEF measures the distance between the clusters using an information theoretic approach. In [9], a generalization of the CEF is defined for multiple clusters. If one associates each of the k labels to a cluster, the CEF can be written as

$$CEF(X') = \frac{1}{2N_1 N_2 ... N_k} \sum_{i=1}^{n} \sum_{j=1}^{n} M(x_i, x_j) G_{\sigma_{l_i}^2 + \sigma_{l_j}^2}(x_i - x_j), \tag{9}$$

where $M(x_i, x_j)$ is a membership function, the value of which is 0 if x_i and x_j have the same label, and 1 if they have different labels.

Equations 8 and 9 together lead to the generalized C-S divergence for multiple labels

$$D_{C-S}(X') = \sum_{l=1}^{k} H(l) - 2 \log CEF(X'). \qquad (10)$$

2.1.1. Computational Complexity of the D_{CS}

In any optimization algorithm based on a utility measure, such as the Differential Evolution (DE), the computational complexity of the utility measure is of great concern.

By close inspection of equations 8, 9 and 10, we can deduce that the computational cost of D_{CS} is quadratic on the number of data points (n).

Let's assume a constant cost for the calculation of $G_\sigma(x_i - x_j)$ (this can be assumed because we know that $x'_i \in \Re^2$). Because of the membership function $M(x_i, x_j)$ in 9, we can calculate the sum of both equations 8 and 9 in one double pass of the observations.

As shown in the Algorithm 1, M implies that when x_i and x_j have the same label k, then the algorithm adds to the calculation of $H(k)$, else the algorithm adds to the calculation of $CEF(X')$.

Algorithm 1 Pseudo code for the calculation of D_{CS}

```
for i = 1 to n do
  for j = 1 to n do
    if label($x_i$) is equal to label($x_j$) then
      H(label($x_i$) = H(label($x_i$) + $G_{2\sigma_l^2}(x_i - x_j)$)
    else
      CEF = CEF + $G_{\sigma_{l_i}^2 + \sigma_{l_j}^2}(x_i - x_j)$,
    end if
  end for
end for
$N = 2$
for i = 1 to k do
  $N = N * n_i$
  $H = H + \frac{1}{n_i^2} H(i)$
end for
$DCS = H - 2 * \log \frac{CEF}{N}$
```

2.2. Differential Evolution

Having defined a quality measure for the class separation of a projection, the next goal is to build a routine to search the projection that maximizes this measure.

In this work, we will search for a linear projection. This linear projection is defined by two parameter vectors A_1 and A_2, where $A_i = a_{i1}, a_{i2}, ..., a_{im} | a_{ij} \in \Re$. The projection function is then defined as $P(x) = [x_i \times A_1, x_i \times A_2]$. Therefore, We want to find A_1 and A_2 that maximize D_{C-S} for the projection.

We use Differential Evolution (DE) [19, 23] to search for these parameter vectors. DE is a simple and powerful populational optimization heuristic inspired by biological evolutionary processes. Its main steps can be briefly described as follows.

DE starts with a set P of p candidate solutions, each generated randomly. Each candidate solution $A_i \in P$ is represented as array of parameters $A_i = a_i 1, a_i 2 ... a_i m \ | a_{ij} \in \Re$, where m is the number of dimensions (attributes) given by the data set. The values of $a_1, ..., a_m$ for each A_i in the initial set P are randomly drawn from a uniform distribution from -1 to 1. In DE literature, the set P is sometimes called a *population*, and each candidate solution A_i an *individual*.

The next step is to iterate this candidate set. At every iteration (sometimes called a *generation*), each candidate $A_i \in P$ is evaluated using the D_{CS}. The resulting utility value is stored as $V(A_i)$.

After the evaluation step, each candidate solution A_i tries to create a new candidate solution A'_i using a procedure called “differential crossover”. This procedure takes three steps. First, three solutions $A_a, A_b \text{and} A_c \in P | a \neq b \neq c$, are randomly selected. Then, a temporary individual A_t is generated as

$$A_t = A_a + F(A_b - A_c), \tag{11}$$

where F is the *differential weight* parameter. The third step is to generate A'_i from A_t and A_i as follows. For each $j \in 1..m$, the value of a'_{ij} in A'_i is taken from either A_t (with probability CR) or from A_i (with probability $1 - CR$). The parameter CR is called the *Crossover Probability.*

After A'_i is generated, its utility value $V(A'_i)$ is calculated. If $V(A'_i) > A(A_i)$, then A'_i replaces A_i in P. Else, it is discarded.

The evaluation and differential crossover step composes one iteration of the DE algorithm. Iterations are repeated until a certain stop criterion, such as a fixed number of iterations or a fixed number of evaluations, is reached. At that moment, the candidate solution in P with the highest utility value is chosen as the projection generated by the algorithm.

By using the differential crossover operator, DE is able to sample the solution space at promising locations. This operator, allied with a large number of solutions in the initial set, allows DE to avoid getting stuck in local optima [15]. This makes DE particularly useful for real-valued, multi modal parameter optimization domains.

In this work, the values used for the DE parameters are: Size of the initial set $p = 50$, $F = 0.8$, $CR = 0.9$, and maximum number of iterations $= 20$. These values follow the suggestions defined by Storn and Price [23].

2.3. Data Transformation

The DE Optimization method generates an array of values in $\Re$. These values are weights for the attributes in the data set, and define a one-dimensional projection of the data set that maximizes the D_{C-S} measure.

In order to acquire a second projection, allowing for a bi-dimensional visualization, we execute the DE optimization of D_{C-S} a second time. Furthermore, we force the second projection to be *orthogonal* to the first. Some ways to do this are described by Zhu [29]. In this work we use a transformation on the data based on the first projection.

To calculate this transformation for solution A given by DE, we find the matrix T such that

$$T = A^t(AA^t)^{-1}A. \tag{12}$$

T is used to transform the original data X

$$Y = (I - T)X, \tag{13}$$

where I is the identity matrix of the same order as T. A second run of the DE algorithm is executed, using Y from eq. 13 as the data set. We take the projection A_2 generated by this run as the second dimension of the desired bi-dimensional projection. When performing the visualization, this second projection must be applied to the transformed data set ($(I - P)X$) to generate the second dimension of the projected data.

3. Experimental Setup

To analyze and validate the performance of the proposed method, we executed two sets of experiments. In the first set of experiments, synthetic data sets were used to test the robustness of the optimization heuristic to its initial conditions and the robustness of the method to very noisy dimensions, respectively. In the second set of experiments, the proposed method was exposed to real world data sets.

3.1. Synthetic data sets

The first data set is composed of 600 $\Re^2$ observations. For each observation, 2 values, x_1 and x_2, were drawn from a uniform distribution between 0 and 1 inclusive. Each observation where $\frac{1}{3} < x_1 < \frac{2}{3}$ was assigned to class 1, and the other observations assigned to class 2. The resulting data set can be seen in Figure 2(a), where the circles represent class 1, and the triangles represent class 2.

The two classes in this data set are in fact separable in just one dimension. The purpose of this experiment was to test how reliably the different methods were able to pick this dimension as the optimal projection.

The following 5 data sets were designed to test the robustness of the methods to noise. Each data set has 600 observations belonging to 3 classes (200 observations for each class). Each data set has 3,4,7,12 and 22 dimensions. For

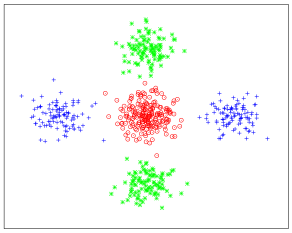


Figure 1: The separable part of the data for experiment 2. To this data, a number of noise dimensions is added. Circles, crosses and stars represent the three different labels.

each data set, the observations were generated as follows: x_1 was drawn from a Normal distribution with standard deviation 1 and mean 0 for classes 1 and 3, and mean -6 or 6 (equal probability) for class 2; x_2 was drawn from a Normal distribution with standard deviation 1 and mean 0 for classes 1 and 2, and mean -6 or 6 (equal probability) for class 3; all the other dimensions $x_i | i > 2$ were drawn from a uniform distribution from -10 to 10.

In other words, the first two dimensions are the same for all sets, and then each set has an increasing number of noise dimensions (1, 2, 5, 10 and 20, respectively). A plot of the first two dimensions can be seen in Figure 1.

3.2. Real world data sets

The third and fourth data sets are the "Pen Digits" and the "Lung Cancer" data sets, taken from the UCI Machine Learning Repository.

The "Pen Digits" data set consists of 16 features (attributes) extracted from samples of hand written digits from 0 to 9. The goal was to recognize the difference between these digits. For ease of visualization, we select three digits at a time. In this paper we report the results for the following digit groups: (0,6,9), (1,3,7), (1,4,7) and (2,5,8). The data set was separated into a training and a validation subset. The projection was generated from the training subset, and then it was applied to the validation subset for analysis of the results.

The "Lung Cancer" data set has 27 observations (removing those with missing attributes) and 56 attributes. They are divided into three classes, representing three different kinds of lung cancer. Because of the low number of observations, all of them were used both for calculating the projection and analyzing the results.

The fifth, sixth and seventh data sets are flow cytometry data sets. These data sets correspond to normal peripheral blood samples containing several subsets of cells, and the goal was to separate different cell types. Each observation corresponds to one cell. All data sets have 10 attributes obtained from the flow cytometry test. These attributes correspond to the expression of 8 different proteins and 2 parameters of light dispersion.

The fifth data set compares monocytes-related dendritic cells, plasmocytoid dendritic cells and B-lymphocytes (classes 1, 2 and 3). These classes have 824, 140 and 259 observations, respectively. Because of the relatively low number of observations in class 2 with relation to the sixth and seventh data sets, all of the observations were used both for generating the projection and visualizing the results.

The sixth data set compares monocytes and neutrophils (classes 1 and 2) These classes have 5582 and 8989 observations, respectively. We took 300 observations from each class (600 total), and used these as the training set. The optimal projection was calculated from the training set. The remaining observations were used as the validation data to display the results.

The seventh data set compares plasmocytoid dendritic cells and neutrophils (classes 1 and 2) These classes have 140 and 8989 observations, respectively. We took all 140 observations from class 1 and 200 observations from class 2 for the training data set. All the observations are used for the validation set.

3.3. Two Methods for Comparison

We compared the results of our method in the described experiments with those of two methods for data visualization: the classical Principal Component Analysis (PCA) [11] and the Zhu-Hastie method (ZH) [30].

To generate two projection dimensions with the PCA, we used the first and the second principal components as the first and the second dimensions in the projection.

The Zhu-Hastie method is a recently proposed method for data visualization based on the maximization of a likelihood measure through an optimization algorithm [30]. This method uses a Log-likelihood Ratio (LR) statistic as its utility function, based on the estimation of density functions using a local likelihood method. To maximize the LR measure, a gradient descent algorithm was used. The stopping criterion was when the distance between the previous projection vector and the current one was equal or lower than 0.05.

3.4. Kernel Width

The LR measure used by the Zhu-Hastie method, and the Dcs measure used in the proposed method both depend on the kernel width parameter. The choice of kernel width has a significant impact on results [13, 24], and is a hard problem in general for divergence measures that compare pdfs.

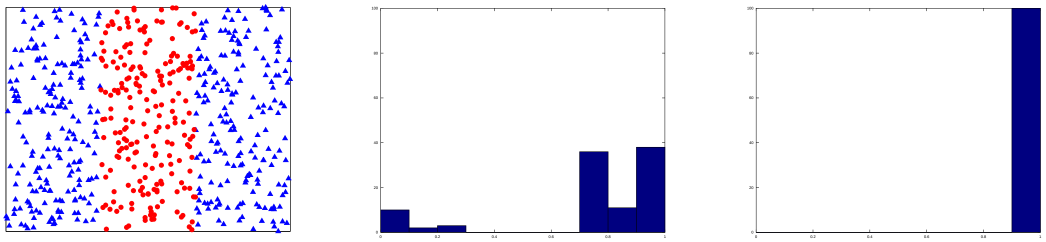
We used a grid search methodology to find the best kernel value for each method. In each experiment, we execute both methods using the following values as the kernel width for each label l: $\{\sigma_l, \sigma_l/20, \sigma_l/40, \sigma_l/60, \sigma_l/80$ and $\sigma_l/100\}$. The best result for each method is reported here (along with its respective kernel value).

As a general trend, we found out that smaller kernel values (larger denominators) produced less reliable results for the proposed method. For the Zhu-Hastie method, on the other hand, we couldn't find such a noticeable trend. Table 1 illustrates this findings.

It is important to note that on Table 1, higher divergence values don't necessarily represent better solutions. While we are interested in maximizing the

Kernel value	Proposed Method	Zhu-Hastie Method
σ	4.98 (4.82, 5,13)	100.99 (89.98, 111.99)
$\sigma/20$	12.71 (11.90, 13.53)	38.70 (37.79, 39.60)
$\sigma/40$	24.81 (20.43, 29.20)	47.95 (47.25, 48.65)
$\sigma/60$	30.65 (24.23, 37.06)	52.74 (52.03, 53.46)
$\sigma/80$	34.17 (28.85, 39.49)	55.49 (54.73, 56.25)
$\sigma/100$	50.60 (40.29, 60.92)	57.45 (56.69, 58.20)

Table 1: Divergence value for the lung cancer experiment. The values within parenthesis are the 95% confidence interval calculated on 50 repetitions.



(a) Data Set 1

(b) Zhu-Hastie

(c) Proposed Method

Figure 2: Results for experiment 1. The histograms in 2(b) and 2(c) show the cosine values between the projections found for each method, and the horizontal axis.

divergence value for a given projection, decreasing the kernel width usually results in an overall increase of divergence values, for both "good" and "bad" projections. It is important to compare these values within a single run of the algorithm.

4. Results

4.1. Synthetic Data Sets

In the first experiment we evaluated the robustness of the optimization heuristic to random initialization. The goal of each run was to find the one-dimensional projection that best separates the two classes in the data set. Figure 2(a) shows the original data. For this data set, the ideal projection is to simply take the value in the horizontal axis.

We executed a hundred runs with the ZH method and with the proposed method. To evaluate the results, we calculated the distance between the projections found by each method and the ideal projection. This distance is calculated as the cosine between the projection and the horizontal axis. An optimal projection would have a cosine value of 1.

Figures 2(b) and 2(c) shows the distances of the projections found by each method as a histogram of cosine values. The ZH method turned out to be quite

Table 2: Results for experiment 1: Cosine between projection and X-axis.

Method	Mean (100 runs)	std
Proposed Method	0.9991	0.0024
Zhu-Hastie	0.7548	0.2933
PCA	0.4532	-

sensitive to initialization, reaching a number of non-optimal solutions. The proposed method was able to find the optimal projection in almost all the runs.

In Table 2 we see the average cosine value for the solutions generated by two methods, and its standard deviation. We also added the cosine value of the projection found by the PCA which does not depend on initial conditions. We can see that the PCA was not able to separate well in this experiment, since the direction with greatest variance of the data set is the diagonal of the rectangle described by the data.

In the second experiment the three methods (the proposed method, PCA, and ZH) were requested to find a bi-dimensional projection for a data set composed of 2 dimensions with actual data (as shown in Figure 1), and 1,2,5,10 or 20 noise dimensions. The results can be seen in Figure 3.

The results can be seen in Figure 3.

For the instance with only 1 noise dimension, the PCA was not able to separate two of the three classes. This is expected, since the variance in the noise dimension is much higher than the variance of the other two dimensions. The ZH Method performed better than the PCA, but two of the three clusters were still very mixed. The proposed method found a bi-dimensional projection of the data with well separated clusters.

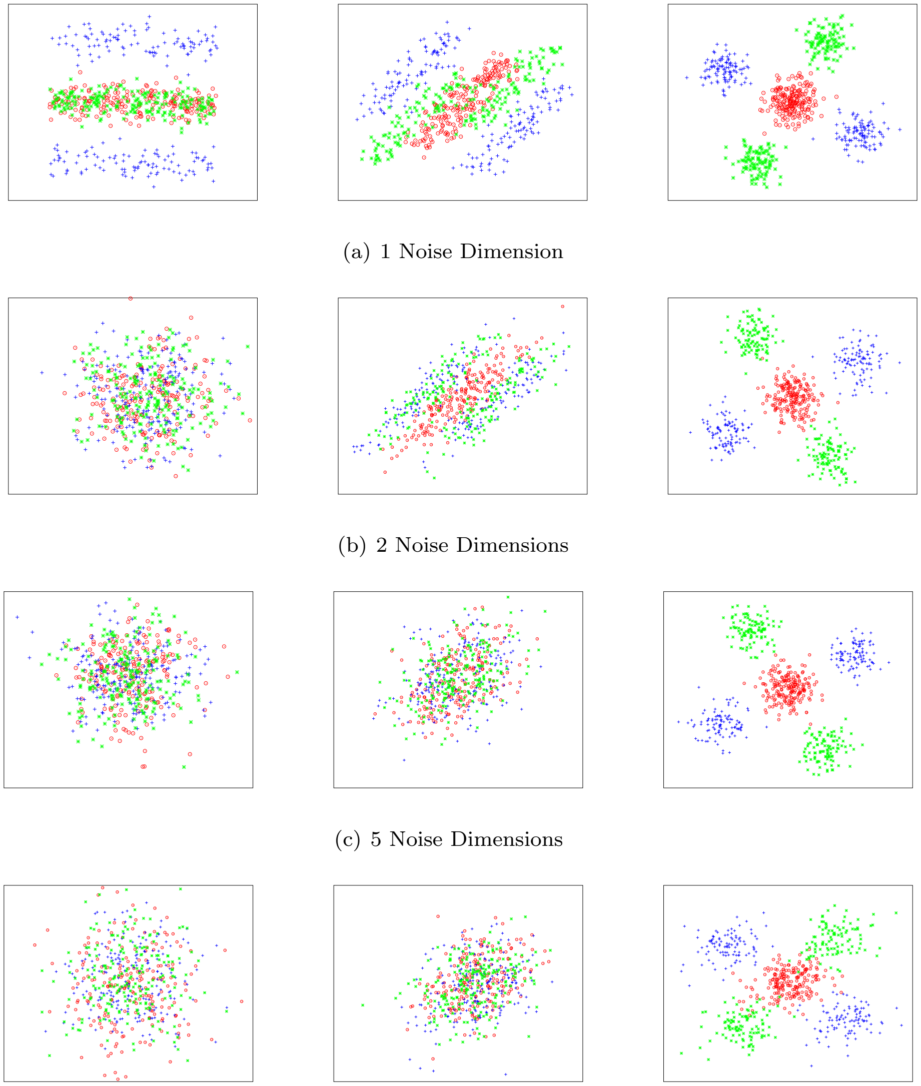
For the instance with 2 noise dimensions, the PCA is no longer able to separate the classes at all. The ZH method also fails to separate the classes, but a sort of “layer” structure is kept. The proposed method still manages to generate a well separated projection.

For 5 and 10 noise dimensions, both the PCA and the ZH are unable to separate the data, and show similar results. The proposed method is still able to separate the classes, but for 10 noise dimensions the clusters start to mix. For 20 noise dimensions, all three methods show the same mixed result (not shown in the figure).

4.2. Real World Data Sets

To examine the performance of the proposed method in practice, we made projections of five data sets based on real problems. For each data set, we visually examined the obtained projections. The resulting projections from each method are shown in Figures 4 to 8.

Figure 4 shows the results for the UCI Pen Digits data set. The results here are not as clear cut as in the artificial data set. In the data sets including the digits 1 and 7, the proposed method has a slightly better separation of these



(a) 1 Noise Dimension

(b) 2 Noise Dimensions

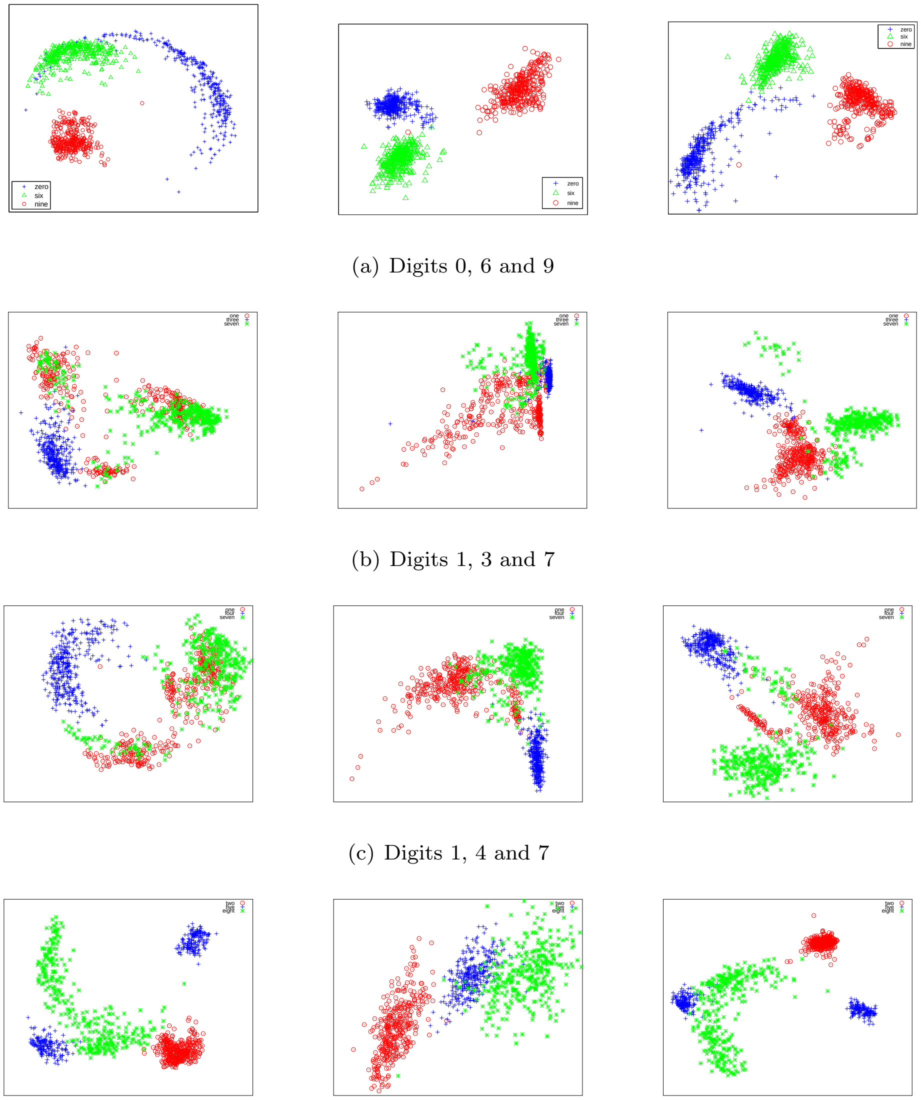
(c) 5 Noise Dimensions

(d) 10 Noise Dimensions

Figure 3: Results for experiment 2. The left column corresponds to the PCA, the middle column to the ZH ($\sigma/40$), and the right column to the proposed method (σ).

two clusters than the ZH and the PCA. In the “0, 3 and 9” and the “2, 5 and 8”, it is harder to tell which is the better projection.

Figure 5 shows the results for the Lung Cancer. Here, the PCA and the ZH methods did not separate the clusters at all (although the PCA roughly ordered the cases by cluster in the X axis). The proposed method managed to separate



(a) Digits 0, 6 and 9

(b) Digits 1, 3 and 7

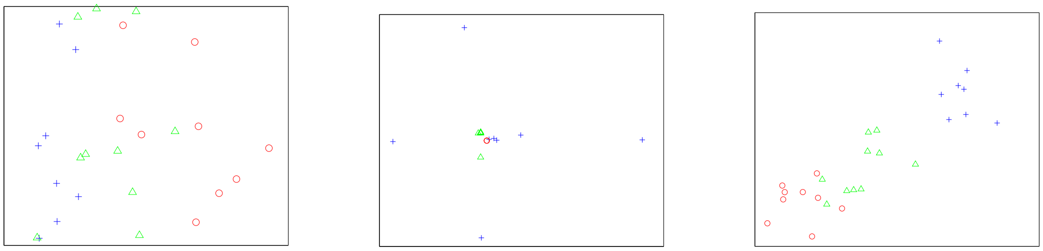
(c) Digits 1, 4 and 7

(d) Digits 2, 5 and 8

Figure 4: Projections found for the UCI Pen Digits data set. The left column corresponds to the PCA, the middle column to the ZH (σ), and the right column to the proposed method ($\sigma/80$).

the classes, but a few elements of class 2 fell in the wrong cluster.

Figures 6, 7, and 8 show the results for the flow cytometry data sets. In the first cytometry data set (Fig. 6), with three classes, the PCA was not able

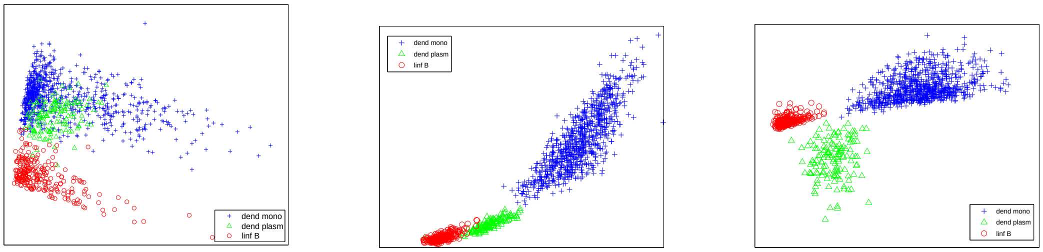


(a) PCA

(b) ZH (kernel=σ)

(c) Proposed Method(kernel=σ)

Figure 5: Projections found for the UCI Lung Cancer data set.



(a) PCA

(b) ZH (kernel=σ)

(c) Proposed Method (kernel=σ/20)

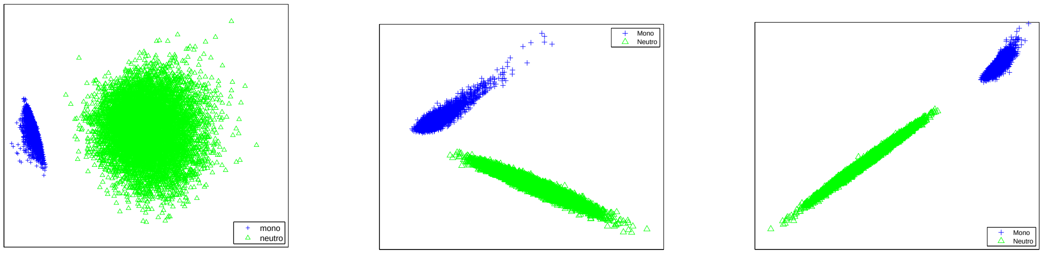
Figure 6: Projections found for monocytes-related dendritic cells (dend mono), plasmocytoid dendritic cells (dend plasm) and B-lymphocytes (linf B).

to separate the monocytes-related dendritic cluster from the plasmocytoid dendritic cluster, and barely managed to separate these two clusters from the B-lymphocytes class. The ZH method generated different clusters for the three classes, but the clusters were too close to each other, and there was a confusion region in the borders. The proposed method, on the other hand, clearly separated the three classes into visually different clusters.

For the other two cytometry data sets (Fig. 7 and 8) all methods created well defined clusters, but the projections of the ZH and the proposed method were more compact than those of the PCA. This is desirable, since it suggests that out-of-sample points are less likely to be placed in the wrong cluster. Also, the distance between the two clusters is much larger in the proposed method than in the ZH or PCA.

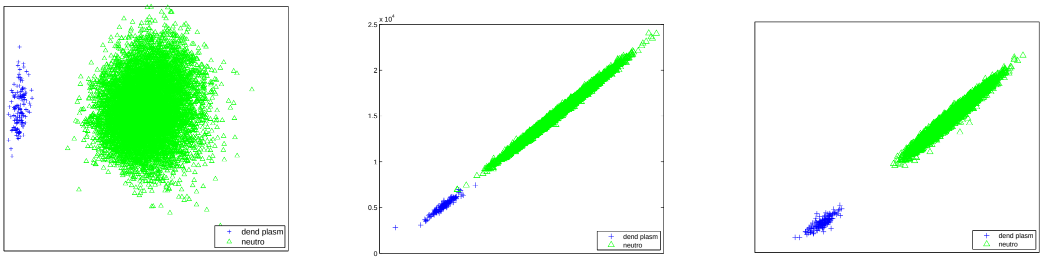
5. Discussion and Conclusion

In this paper, we have introduced a new system for the visualization of multidimensional data as a bi dimensional image. This system uses an extension of



(a) PCA (b) ZH (kernel=$\sigma/100$) (c) Proposed Method (kernel=σ)

Figure 7: Projections found for monocytes-related dendritic cells (mono) and neutrophils (neutro)



(a) PCA (b) ZH (kernel=σ) (c) Proposed Method (kernel=σ)

Figure 8: Projections found for plasmocytoid dendritic cells (dend plasm) and neutrophils (neutrofilos).

the Cauchy-Schwartz divergence measure for multiple clusters as the measure of the projection's quality, and Differential Evolution to generate a projection function that maximizes this measure. Using this method, we promote the bidimensional visualization of high-dimensional data sets with optimized cluster separation.

Using experiments with synthetic data, we have shown that our method is generally robust, specially regarding the addition of noise to the data set. We compared this result with the PCA, a well established and broadly used method for data visualization, and a recently proposed method that also uses a divergence measure as its metric.

It is worth mentioning that very good results were recently published using the PCA for flow cytometry data [2]. The proposed method produced clearly better results than the PCA for some of the cytometry data sets (Figs. 6(a) and 6(c)). This shows a promising perspective concerning its application in this area.

On the experiments with real world data sets, the proposed method was generally competent at generating informative projections of the data. For these experiments, it is harder to set an objective measure of cluster separation: each method optimizes a different divergence measure, so their value can't be used to compare the methods directly.

Still, we can observe that the Cauchy-Schwartz divergence measure was able to define projections where the clusters are well separated. The sum of the quadratic densities of the labels promotes compact clusters, while the use of the Cluster Evaluation Function (CEF) promotes the separation between them.

Differential Evolution showed to be a good method for optimizing a projection based on D_{C-S} as a utility measure. Further improvement can probably be achieved by the use of more recent DE techniques such as [26].

5.1. Future Works

The findings on this work suggest a number of ways that research can take place in order to improve available solutions to the Data Visualization problem.

Firstly, there is a concern regarding the time complexity of the D_{CS}. While the method as is has performed well on current problems, it may be slower to train on data sets with a larger number of cases. However, population-based meta-heuristic optimization methods, such as DE, are "embarrassingly parallel" algorithms. We are currently concentrating our efforts on the development of a parallel version of the DE-DCS system.

Also, we observe that the two parts of the D_{C-S}, the CEF and the Quadratic Density, measure essentially different things. Because of this, it would be interesting to try and treat them as different objectives in a multi-objective optimization system. This would generate a Pareto set of solutions, allowing for the selection of the most appropriate ones, depending on the priorities for the visualization of a particular data set.

Another interesting topic is how to evaluate the quality of bi-dimensional data projections for human consumption. For two different divergence criteria, how to determine which one, when optimized, produces the better image? In this case "better" is heavily dependent on the context of the problem being approached, and a careful selection of human judges or comparison metrics is necessary.

Acknowledgment

Carlos E. Pedreira was partially supported by research grants from CNPq (Brazilian National Research Council) and FAPERJ (Rio de Janeiro Research Foundation). Claus Aranha was supported by a Pst-Doc grant from CNPq. Rodrigo Peres was supported by a Post-Doc grant from FAPERJ/CAPES.

The authors would like to thank Kathleen Gonçalves and Elaine S. Costa for the flow cytometry data sets used in this research and also for a number of suggestions that greatly improved this manuscript.

The authors would also like to thank Juliana A. Freire for reviewing the text.

The authors acknowledge the UCI Machine Learning Repository for the "Pen-Based Recognition of Handwritten Digits" and "Lung Cancer" databases used in this research.

References

[1] D. Asimov, The grand tour: A tool for viewing multidimensional data, SIAM Journal of Science & Stat. Comp. 6 (1985) 128–143.

[2] E.S. Costa, C.E. Pedreira, S. Barrena, Q. Lecrevisse, J. Flores, S. Quijano, J. Almeida, M. del Carmen Garcia-Macias, S. Bottcher, J.J.M.V. Dogen, A. Orfao, on behalf of the EuroFlow Consortium, Automated pattern-guided principal component analysis vs expert-based immunophenotypic classification of b-cell chronic lymphoproliferative disorders: a step forward in the standardization of clinical immunophenotyping, Leukemia 24 (2010) 1927–1933.

[3] E.S. Costa, R.T. Peres, J. Almeida, Q. Lecrevisse, M.E. Arroyo, C. Teodosio, C. Pedreira, J.J.M.V. Dongen, A. Orfao, on behalf of the EuroFlow Consortium, Harmonization of light scatter and fluorescence flow cytometry profiles from intracellular stainings, Cytometry Part B - Clinical Cytometry 78b (2010) 11–20.

[4] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley and Sons, 2006.

[5] P.L. D. Koloseni, J. Lampinen, Optimized distance metrics for differential evolution based nearest prototype classifier, Expert Systems with Applications 39 (2012) 10564–10570.

[6] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Transactions on Systems, Man and Cybernetics, Part A 38 (2008) 218–237.

[7] R. Duda, P. Hart, G. Stork, Pattern Recognition, Wiley, 2nd edition, 2001.

[8] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936).

[9] E. Gokcay, J.C. Principe, Information theoretic clustering, Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 158–171.

[10] T. Hastie, R. Tibshirani, J. Friedman, The elements of statistical learning, The Elements of Statistical Learning, Springer, 2001.

[11] I.T. Jolliffe, Principal Component Analysis, SpringerVerlag, 1986.

[12] P. Luukka, J. Lampinen, Differential evolution classifier in noisy settings and with interacting variables, Applied Soft Computing 11 (2011) 891–899.

[13] S. Maldonado, R. Weber, J. Basak, Simultaneous feature selection and classification using kernel-penalized support vector machines, Information Sciences 181 (2011) 115 – 128.

[14] U. Maulik, I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recognition 42 (2009) 2135–2149.

[15] N. Noman, H. Iba, Accelerating differential evolution using and adaptive local search, IEEE Transactions on Evolutionary Computing 12 (2008) 107–125.

[16] A. Orfao, Useful information provided by the flow cytometric immunophenotyping of hematological malignancies: Current status and future directions, Clinical Chemistry 45 (1999) 1708–1717.

[17] C.E. Pedreira, E.S. Costa, J. Almeida, C. Fernandez, S. Quijano, J. Flores, S. Barrena, Q. Lecrevisse, J.J. van Dongen, A. Orfao, on behalf of the EuroFlow Consortium, A probabilistic approach for the evaluation of minimal residual disease by multiparameter flow cytometry in leukemic b-cell chronic lymphoproliferative disorders, Citometry A 12 (2008) 1141–1150.

[18] C.E. Pedreira, E.S. Costa, M.E. Arroyo, J. Almeida, A. Orfao, A multidimensional classification approach for the automated analysis of flow cytometry data, IEEE Transactions on Biomedical Engineering 55 (2008) 1155–1162.

[19] K. Price, R. Storn, J. Lampinen, Differential Evolution - A Practical Approach to Global Optimization, Springer, 2005.

[20] J.C. Principe, D. Xu, J. Fisher, Information theoretic learning, in: S. Haykin (Ed.), Unsupervised Adaptive Filtering, Wiley, 2000.

[21] R.Jensen, An Information Theoretic Approach to Machine Learning, Ph.D. thesis, Faculty of Science, Department of Physics, University of Tromso, Tromso, Norway, 2005.

[22] R. Shepard, The analysis of proximities: multidimensional scaling with an unknown distance function, Psychometrika 27 (1962) 125–140.

[23] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[24] A. Unler, A. Murat, R.B. Chinnam, mr2pso: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Information Sciences 181 (2011) 4625 – 4641.

[25] S. Vinga, J. Almeida, Rényi continuous entropy of dna sequences, Journal of Theoretical Biology 231 (2004) 377–388.

[26] Y. Wang, Z. Cai, Q. Zhang, Enhancing the search ability of differential evolution through orthogonal crossover, Information Sciences 185 (2012) 153 – 177.

[27] D.M. Witten, R. Tibshirani, Supervised multidimensional scaling for visualization, classification and bipartite ranking, Computational Statistics and Data Analysis 55 (2011) 789–801.

[28] D. Xu, Energy, Entropy and Information Potential for Neural Computation, Ph.D. thesis, University of Florida, Gainesville, FL, USA, 1999.

[29] M. Zhu, Feature Extraction and Dimension Reduction with Applications to Classification and the Analysis of Co-occurrence Data, Ph.D. thesis, Stanford University, 2001.

[30] M. Zhu, T.J. Hastie, Feature extraction for nonparametric discriminant analysis, Journal of Computational and Graphical Statistics 12 (2003) 101–120.

Vitae

Rodrigo T. Peres was born on 25th August 1976 in Rio de Janeiro, Brazil. He received Bachelor's degree in Mathematics from Universidade Federal Fluminense, UFF, Rio de Janeiro, Brazil, in 2000, MSc and PhD degrees in electrical engineering from PUC-Rio, Rio de Janeiro, Brazil, in 2004 and 2008, respectively. He is currently a Post-Doc fellow at the School of Medicine and COPPE-PEE - Engineering Graduate Program at Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil.

Claus C. Aranha received a Bachelor's and MSc degrees in computer sciences from the State University of Campinas, Brazil, in 2001 and 2005, respectively. He received MSc and PhD degrees in Frontier Informatics from the University of Tokyo, Japan, in 2007 and 2010, respectively. He is currently an assistant professor at the graduate school of Systems and Information Engineering at the University of Tsukuba. His main research interests include: Application of Evolutionary Computation, Parallel Evolutionary Algorithms and Bio informatics.

Carlos Eduardo Pedreira received his Bachelor's and MSc degrees in electrical engineering from the Catholic University of Rio de Janeiro in 1979 and 1981, respectively, and a PhD degree from the Imperial College of Science Technology and Medicine, University of London, UK, in 1987. He is currently aProfessor at the COPPE-PEE - Engineering Graduate Program and at the School of Medicine at Federal University of Rio de Janeiro (UFRJ), Brazil. He was the Founding President of the Brazilian Neural Networks Society. He is a member of the EuroFlow consortium. His main research interests include Pattern classification, Cluster analysis and Flow cytometry data.