An Efficient Optimization Method for Revealing Local Optima of Projection Pursuit Indices

Souad Larabi Marie-Sainte[1], Alain Berro[1], and Anne Ruiz-Gazen[2]

[1] IRIT-UT1, UMR 5505, CNRS;
Université Toulouse 1 - Capitole
{souad.larabi,alain.berro}@irit.fr
[2] Toulouse School of Economics (GREMAQ);
Université Toulouse 1 - Capitole
ruiz@cict.fr
2, rue du doyen Gabriel Marty, 31042 Toulouse Cedex, France

Abstract. In order to summarize and represent graphically multidimensional data in statistics, projection pursuit methods look for projection axes which reveal structures, such as possible groups or outliers, by optimizing a function called projection index. To determine these possible interesting structures, it is necessary to choose an optimization method capable to find not only the global optimum of the projection index but also the local optima susceptible to reveal these structures. For this purpose, we suggest a metaheuristic which does not ask for many parameters to settle and which provokes premature convergence to local optima. This method called Tribes is a hybrid Particle Swarm Optimization method (PSO) based on a stochastic optimization technique developed in [2]. The computation is fast even for big volumes of data so that the use of the method in the field of projection pursuit fulfills the statistician expectations.

1 Introduction

Exploratory projection pursuit techniques aim to reveal visually an interesting structure hidden within multivariate data ([9], [10], [1]). This family of statistical methods consists in detecting interesting linear projections by optimizing a predetermined function called projection index that measures in some sense the "interestingness" of a projection.

The projection pursuit is based on two important elements: the projection index and the optimization algorithm. The literature exposes several projection indices and optimization methods. These methods are global optimization methods such as the gradient's method [10], the ascent's method ([7], [13], [14], [15]), the quasi-Newton's method ([7], [15]) and some modified version of Newton's method [14]. Several projections of the data may reveal interesting structures. So, in order to obtain different local optima, the aforementioned algorithms work in the following way. They look for a global optimum of the projection index and when a solution is found, it is removed from the space of solutions for instance by projecting the data in the orthogonal space of the global solution. Then, the index is optimized again in order to find other solutions. Several projections (local optima) in the initial search space may not be detected when we consider the successive orthogonal spaces. Furthermore, the optimization methods quoted above require the calculation of the gradient and may ask for a meticulous choice of an initial point (initial solution). Our objective is not only to find the global optimum of the index but also the local optima to reveal these possible interesting structures. We also wish to suggest to the statistician an algorithm without parameter tuning and which enough explores the space to find various local optima of the index without considering orthogonal spaces. Furthermore, the problem of the optimization of these projection indices is complex and expensive in computing time and the solutions proposed to maximize (minimize) these indices are not always convincing. Therefore, the projections pursuit methods are little used and absent from the well known statistical softwares (except Matlab [12] or quasi-clones, like SciLab and Octave, and GGobi [5]). Our purpose is to propose powerful and fast algorithm allowing the detection of several local optima.

The Particle Swarm optimization (PSO) and Tribes are metaheuristics that appeared recently. They differ from the other evolutionary methods (typically, the genetic algorithms) and are based on the notion of cooperation between agents (particles). The information exchanged between particles gets to resolve difficult problems. These techniques present some interesting peculiarities, among others, the notion of efficiency due to the collaboration rather than the competition. Furthermore, the fact that these methods converge early to local optima is an interesting feature in order to find new potentially interesting projections. In a first work, we used the PSO and the genetic algorithms to optimize certain projection indices and both optimization methods have proven their efficiency but they need parameters to be tuned. Note that PSO is also used for Projection Pursuit in [16] in the context of regression.

Contrary to PSO, Tribes is presented as a black box, because it possesses no parameter to settle and it easily exhibits satisfactory performances. In this technique, particles are divided into several tribes or groups of variable size. Tribes method presents the risk of a too fast convergence, which can be translated by the fact that it finds local optima. To remedy this problem, Clerc ([3], [4]) proposed a new version of Tribes. As far as our objective is not only to find the global optimum but also several potential local optima, we prefer to apply the native version. Tribes was never used in the field of projections pursuit. Its application in this article shows that it can lead to better results than the classical PSO as shown in the previous work.

In this paper, we present a comparison of the Tribes technique with the classical PSO version applied to the exploratory projections pursuit to optimize one-dimensional projection index. We focus on the search of clusters among any other interesting structure such as outliers. In section 2, we introduce briefly the problem of projection pursuit and the two projection indices we focus on. Section 3 presents the technique of particle swarm optimization briefly and the technique of Tribes in more detail. The last section is dedicated to the comparison of these two techniques on some small data sets.

2 Exploratory Projection Pursuit

The Exploratory Projection Pursuit (EPP) techniques consist in the search for hidden aspects within a big volume of data [8]. The objective of these exploratory techniques is to look for low (one, two or three) dimensional projections that provide the most revealing views of the full-dimensional data. The search for such projections requires the definition of a numerical index $I(a)$ for every projection a. The intent of this index is to capture nonlinear structures present in the distribution of the projected data. This function is defined so that the interesting projections correspond to the global optimum and to the local optima of this function.

Principal components analysis (PCA) is a familiar exploratory technique of this kind, it is a projection pursuit method where the index of interestingness represents the variance of the projected data. Its efficiency has been relativized [10] because certain important projections may not appear in the principal subspaces, even if their dimension is small. Furthermore, the maximization of this index (the variance) can be solved by using the spectral decomposition so that PCA does not need any optimization algorithm.

As in many situations of data analysis, we consider N individuals characterized by P variables. To every individual corresponds a vector X_i in $\mathbb{R}^P (i = 1, \cdots, N)$ which is assimilated to a matrix column, the transposed of these vectors leads to a matrix X with dimension $N \times P$. Unfortunately, it is not possible to visualize points in P-dimensional space if P is upper to 3. However, it is possible to project a P-dimensional set of points onto a one-dimensional line. The projection is a linear function of $\mathbb{R}^P$ towards $\mathbb{R}$ of N observations $X_1, \cdots, X_N$ such as $z = Xa$. The P-vector a defines a linear transformation and the N column-vector z corresponds to the projected data coordinates. The problem consists in determining a projection a. As usual in EPP, we suppose that the data are spherical (by transforming the data accordingly), such that the mean vector $E(X_i) = 0$ and the covariance matrix $V(X_i) = I_P$ where I_P denote the identity P-dimensional matrix. By considering spherical data, PP is going beyond the first and second moments of the data which are already taken into account in standard analysis such as Principal Components analysis.

There are many possible projection indices, the present paper focus on a one dimensional polynomial-based index named the Friedman index [7] and a moment-based index called the kurtosis index [14]. We limit ourselves to a brief definition of these two indices.

2.1 The Friedman index

This index is based on the Legendre polynomials [7]. It measures the departure between the density of the projected data and the normal density which is assumed to correspond to a non-interesting projection. The formula is given as follows:

$$I_h^F(a) = \sum_{j=1}^{h} \frac{2j+1}{2} \left[\frac{1}{N} \sum_{i=1}^{N} L_j\{2\Phi(X_i) - 1\} \right]^2 \quad (1)$$

where Φ is the univariate standard normal distribution. The recursive definition of the Legendre polynomials is given by:

$$\begin{array}{l} L_0(r) = 1, L_1(r) = r, L_2(r) = \frac{1}{2}(3r^2 - 1), \\ L_j(r) = \frac{1}{j}(2j-1)rL_{j-1}(r) - (j-1)L_{j-2}(r) \text{ pour } j \geq 3 \end{array} \quad (2)$$

The choice of the value of h depends on the data dimension P and the sample size N. In the present article h is fixed to 3 according to the recommendations given in [7] and [15].

2.2 The kurtosis index

This index is based on the fourth moment of the projected data distribution [14]. It is the kurtosis coefficient of the projected data. The directions are chosen by minimizing and maximizing this coefficient. The minimization of the kurtosis coefficient implies the maximization of the "bimodality" of the projections, that leads to the determination of clusters, whereas its maximization leads to the detection of outliers [14]. The index is defined as follows:

$$I_k(a) = \sum_{i=1}^{N} (a^T X_i)^4 \quad (3)$$

3 Bio-inspired algorithms

The optimization algorithm is an important choice in the projection pursuit problem. It consists in finding the directions which maximize (or minimize) the projection index I. This section presents the PSO and Tribes which are bio-inspired algorithms. In other words, there are iterative stochastic methods for global optimization which are inspired by the theory of the biological populations evolution. One of the interests to study these approaches is to develop an algorithm with powerful ability to find out the global and the local optima of the optimization problem. These methods develop a set of solutions with the purpose to find the best results.

3.1 Particle Swarm Optimization (PSO)

The PSO algorithm is an optimization metaheuristic method, invented by Eberhart and Kennedy in 1995 [11]. This method incorporates concepts that lead particles to converge gradually to a local optimum. The PSO algorithm is initialized with a swarm of random candidate solutions, called particles. All the particles have fitness values which are evaluated by the fitness function to be optimized, and are assigned a randomized velocity at the beginning of optimization and are iteratively moved through the problem's searching space. Each particle tries to improve its performance according to its own experience and the experience of its environment.

If $X_m(t)$ represents the position of the particle m to the iteration t, then its velocity at iteration $t+1$ is defined by:

$$V_m(t+1) = w * V_m(t) + r1 * (X_m^* - X_m(t)) + r2 * (X^* - X_m(t)) \quad (4)$$

where $V_m(t)$ is the velocity at the preceding iteration, w is the inertia weight employed to adjust the influence of the previous particle velocities on the optimization process. X_m^* is the best historical position ever obtained by m, X^* is the best particle ever obtained during the algorithm, $r1$ and $r2$ are fixed parameters. We define the new position of the particle m as follows:

$$X_m(t+1) = X_m(t) + V_m(t+1) \quad (5)$$

In our work, the projection index represents the fitness function and the vector of projection defines a particle. In the first work, we used the classic version of the PSO with a modification of the notion of neighborhood in order to adapt the method to EPP. So, the practical application of the algorithm involves using X_l^* being the best particle in the neighborhood instead of X^*.

3.2 Tribes

Tribes is a hybrid PSO method based on a technique of stochastic optimization developed in [2] (see also [4]). This technique is a competitive algorithm which allows to find quickly local optima by investigating simultaneously several regions of the search space, generally local optima, before making a global decision.

In Tribes, particles are divided into several tribes, a metaphor for different sized groups of particles moving about in an unknown environment, looking for a "good" place. Each particle is evaluated by the fitness function (the projection pursuit index I). In each tribe, information links build a completely connected graph. Between tribes, links are looser, but the whole graph is still connected. This graph forms a structure able to diffuse and exploit information. This structure must be automatically generated and updated by means of creation, evolution and deletion of particles and tribes. Moving strategies of a particle, which indicate how a particle must modify its position, are based on "hyperspherical" probability distributions, which may be with or without noise, or independent Gaussian. The choice of these strategies is made depending on the short term history of the particle. This structuring will automatically induce the same purpose, namely explore several promising areas simultaneously, usually around local optima. In this part, we are going to define some notions allowing to understand the mechanism of this technique and give an algorithm describing its complete progress. Let us note that a particle is always defined as being a vector of projection.

The algorithm begins with one particle in a single tribe. Then it consists in creating and deleting particles and tribes. Along iterations, the position of the particles (value of the projection vector) is updated according to some strategies of displacement. Each time a tribe is created, links between particles are defined in order to make possible the transfer of information between tribes (in particular the best position of the particles in each tribe). Creating and deleting particles and tribes rely on measures of quality.

x: *is the best position memorized by the particle during its course*
p: *is the best position memorized by the best particle of the generating tribe*
g: *is the best position memorized by the best particle of the swarm*
$nb\_iteration$: *the number of current iteration*
$Max\_iteration$: *The maximum number of iterations*
L: *the total number of information links*
$nb\_iteration = 0$; $L = 0$;

1. Create a first tribe formed of a single free particle
2. Estimate its fitness (the projection index)
3. Calculate $x = p = g$
4. $nb\_iteration + +$
5. Create the second tribe, from the first tribe,
   consisted of a couple of free and stuffy particles
6. $L = 1$

for $nb\_iteration = 1$ to $Max\_iteration$ **do**
  Estimate fitness of every particle
  Calculate x, p, g for each particle
  Determine the quality of every particle and every tribe
  if Number of tribes < 3 and Both tribes do not improve their performance **then**
    Create the third tribe, from the first two tribes,
    formed of two pairs of free and stuffy particles
    $L + +$
  else
    if $nb\_iteration = \frac{L}{2}$ **then**
      /* Do an adaptation */
      Create a new tribe
      Remove the worst monotribe of the swarm
      Remove the worst particle of each "good" tribe
      $L + +$
    end if
  end if
end for

Algorithm 1: The Tribes algorithm

A particle is characterized by four possible qualities. It is labeled "good" if it has just improved its best performance (fitness value), "neutral" otherwise.

A particle having the least good performance within its tribe is said "worse", it is said "excellent" if its two last variations of performance (between successive iterations) are improvements.

The tribes themselves also receive the labels "good" or "bad", depending on the number of good particles in the tribe. A tribe containing T particles is itself "good" only if $U() \leq G/T$, where G is the number of good particles in a tribe and $U()$ is drawn from a standard uniform distribution. Otherwise the tribe is "bad". Good tribes, because they are doing well and presumably do not need as many particles, will remove one of their particles and only the worst of them, i.e. the particle with the highest value of I, if we assume that the projection index I is the function being minimized. When this occurs, any external links to the particle are re-assigned to the best performer in the tribe, i.e. the particle with the lowest value of I. In the case of a monoparticle tribe, the tribe itself is removed only if we are certain to keep in contact with all the tribes, i.e. all external links to the particle are reassigned to the external best particles. Bad tribes, on the other hand, presumably need more information, so each creates two new particles outside of its tribe and forms a link between the new particles and the best particle within the tribe. The set of all new particles created by all the bad tribes during one adaptation step forms a new tribe.

An adaptation is the realization of a deletion and/or a creation of particles, as described above. It occurs once at the beginning of the algorithm and then periodically as it progresses in order to propagate the information between particles. If, after adaptation, the number of links in the swarm is L, then adaptation will occur again after $L/2$ swarm iterations.

A particle adopts a strategy of movement according to its recent past and which looks like a local search. Three possibilities of variation of a particle's performance exist: deterioration (-), status quo $(=)$ and improvement $(+)$. The confinement of the particle in the search space is realized in the same way as in PSO but without velocity. Because the history of a particle includes two variations of performance, we find 9 possibilities of variation grouped in 3 strategies of movement according to the recommendations of Clerc [2]. The strategies are the pivot if the history of performance is $(--)$ or $(=-)$ or $(-=)$ or $(==)$, the disturbed pivot if the history is $(+=)$ or $(-+)$ or $(+-)$ and the local by independent gaussian if the history is $(=+)$ or $(++)$.

Pivot strategy: the new position of the particle is chosen at random according to an isotropic distribution centred on the pivot, for example a gaussian distribution.

$$x_d \leftarrow C_2 * \text{alea}(H_p) + C_3 * \text{alea}(H_g) \tag{6}$$

with p the best position memorized by the particle in the course of movement, g the best position stored by the best particle of the swarm, H_p, H_g two hyperspheres centred on p and g respectively and of the same radius equal to the distance $||g-p||$, $C_2 = \frac{I(p)}{I(p)+I(g)}$, $C_3 = \frac{I(g)}{I(p)+I(g)}$ and I the projection index

Disturbed Pivot strategy: it is the same strategy as the previous one but with a noise. When we determine the new position, we modify it again according to a random gaussian noise. This noise will be very low if the performance of the particle is good. For every dimension of the space, we have:

$$\begin{cases} \sigma = \dfrac{I(p) - I(g)}{I(p) + I(g)} \\ b_d = \mathcal{N}_d((0, \sigma) \\ x_d = (1 + b_d)x_d \end{cases} \tag{7}$$

Local by independent gaussian strategies: the idea is to look for locally and only around the best position g known by the best particle. So, for every dimension d of the space, a coordinate close to the coordinate g_d of g is chosen at random according to a Normal law

$$x_d \leftarrow g_d + \mathcal{N}_d(0, |g_d - x_d|) \tag{8}$$

After the first iteration, if the situation does not improve, two particles will be generated, forming a second tribe. One of its two particles, called free, is generated anywhere in the search space according to a uniform distribution in the whole space and the other, said stuffy, is uniformly generated in a D-sphere of center g and of radius $||g - x||$, where g is the best position stored by the best particle of the swarm and x is the best position memorized by the best particle of the generating tribe. The idea of this generation is to intensify the search in a region which seems already interesting. At the next iteration, if neither of the two tribes improves its situation, each of the two tribes will generate another couple of new particles simultaneously, forming a new tribe of four particles, and the process will continue as described in the following algorithm. We note that as the number of links increases, the importance of the number of iterations between the two adaptations increases. Between two adaptations, the swarm then has more and more chances to find a solution.

The Tribes method is very useful in the resolution of the PP problem. This technique is efficient in most of the cases and allows the statistician to gain time by avoiding the tuning stage of the metaheuristic. Indeed, the statistician has to supply only the stopping time criterion and the objective function. Furthermore, [4] and [3] indicate that because the parameters are not assigned to their optimal values, the method converges early, resulting in being local optima on certain problems. Tribes is thus a very promising tool for the determination of several local optima which can reveal potentially interesting projections.

4 Application

Clustering is a set of statistical methods that separate data into classes (clusters) but, it is not clear how to assert that a data set contains well defined clusters. Our objective is to detect the presence of potential clusters in multidimensional data by using exploratory projections pursuit methods. We consider the two projection indices defined in section 2. In the present section, we give some results of the PSO and Tribes optimization methods applied to four data sets and demonstrate the interest to apply Tribes for the determination of the local optima of projection pursuit indices. The algorithms of these methods are implemented in language Java.

At first, we specify the number of particles for the PSO and the number of iterations for the PSO and Tribes. As it was recommended by Clerc [2], the PSO needs no more than 50 particles for small data sets. As regards to the number of iterations, we fixed it to 100 for the simulated and olive oil data for both methods. This value has been obtained by carrying out some preliminary runs on each data set and checking the convergence of the indices to local optima. We ran 100 times each optimization algorithm on the different data sets and we present below some of the obtained results. In order to summarize the results in an efficient way, we draw the ranked values of the indices to each of the one hundred local optima with the projection vector corresponding to the best value of the index.

We present plots of the ranked values of the projection indices for the data sets using PSO and Tribes. We note that the number of launches can be increased during the exploration of big volumes of data. We also plot some histograms of the distributions of the projected data associated with local optima of the different indexes in order to visualize possible structure(s).

4.1 Simulated data

We generated three data sets of $N = 1000$ observations and $P = 5$ variables. The observations are distributed according to various mixtures of standard normal distribution indicated as follows:

Normal2 contains two clusters of 500 observations with gaussian distribution $\mathcal{N}_5(\mu_i, I_5)$ with $i = 1, 2$ where $\mu_1 = (0, ..., 0)^T$ and $\mu_2 = (10, 0, ..., 0)^T$ are 5-dimensional vectors.

Normal4 contains four clusters of 250 observations with gaussian distribution $\mathcal{N}_5(\mu_i, I_5)$ with $i = 1, ..., 4$ where $\mu_1 = (0, ..., 0)^T$, $\mu_2 = (10, 0, ..., 0)^T$, $\mu_3 = (0, 10, 0, 0, 0)^T$, $\mu_4 = (0, 0, 10, 0, 0)^T$ are 5-dimensional vectors.

Normal10 contains ten clusters of 100 observations with gaussian distribution $\mathcal{N}_5(\mu_i, I_5)$ with $i = 1, ..., 10$ where $\mu_1 = (0, ..., 0)^T$, $\mu_2 = (10, 0, ..., 0)^T$, $\mu_3 = (0, 10, ..., 0)^T$, $\mu_4 = (0, 0, 10, 0, 0)^T$, $\mu_5 = (0, ..., 0, 10)^T$, $\mu_6 = -\mu_1$, $\mu_7 = -\mu_2$, $\mu_8 = -\mu_3$, $\mu_9 = -\mu_4$, $\mu_{10} = -\mu_5$ are 5-dimensional vectors and I_5 is the identity matrix.

The purpose of this example is to show the efficiency of the Tribes method in the detection of local optima which correspond to projections revealing the clusters structures of the data sets. On figure 1 we plot the 100 ranked values of the minimum kurtosis index for the simulated data with PSO (right curves) and Tribes (left curves). While the PSO method leads to small variability of the projection index values for the one hundred launches, Tribes supplies different

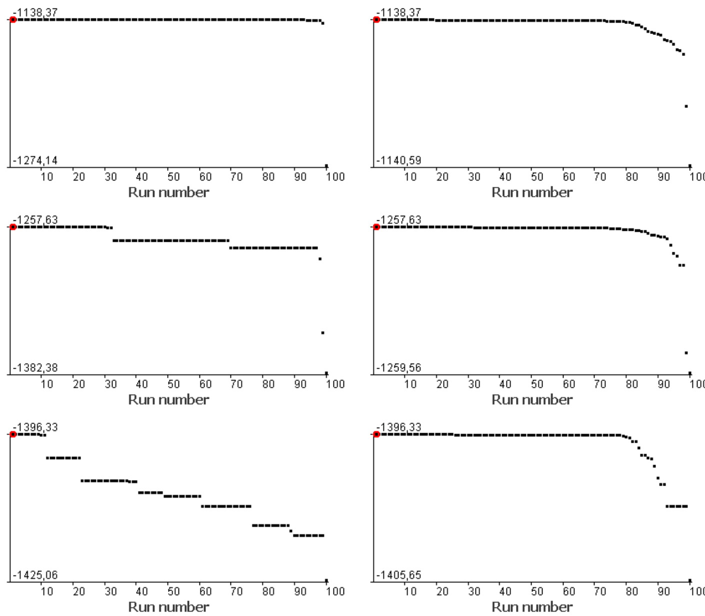

Fig. 1. Simulated data: Plots of the ranked values of the kurtosis index for the Normal2 (top curves), the Normal4 (middle curves) and the Normal10 (bottom curves) with PSO (right curves) and Tribes (left curves).

local optima as soon as the interesting structure is complex and cannot be visualized on one dimension (Normal4 or Normal10). For the first two plots at the top, we tested the first data set Normal2 which contains two clusters. There is an unique interesting structure associated with an optimum index detected by the two methods. Both plots at the middle correspond to the data Normal4 which contain four clusters. We don't give the plots of the projections of the data but the structure in four clusters is detected by looking at the projections associated with the local optima corresponding to the three landings of Tribes. On the contrary, the PSO method does not allow to detect the four clusters. The index values associated to the last data set Normal10, which contains 10 clusters, are represented in the last two plots below. We notice that Tribes proposes several different local optima (see the landings) which represent various interesting projections. For these particular examples, the Friedman index gives the same results as the kurtosis index using both optimization methods.

4.2 Olive data

The file consists in the percentage composition of $P = 8$ fatty acids found in the lipid fraction of $N = 572$ Italian olive oils. The 572 samples come from three different Italian regions subdivided themselves into nine areas. This data set has been analyzed by several authors in the context of exploratory multivariate analysis (see [6] and [1]). The structure of the data set is quite complex with nine clusters which have different shapes in an eight-dimensional space. Due to the large number of classes, discovering all of them by using one-dimensional EPP is challenging but the results we obtain clearly highlight a complex groups structure since several groups are detected by processing the data with the two proposed indices using Tribes.

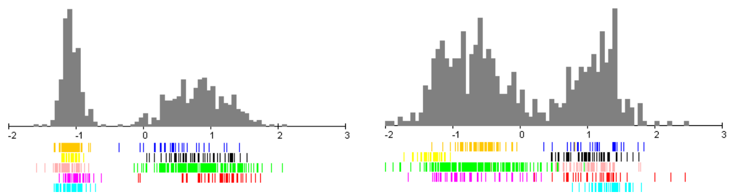

Fig. 2. Olive data: histogram corresponding to the global optimum (left figure) and a local optimum (right figure) for the minimum kurtosis index using Tribes.

As for the simulated examples, the plots (not given in the present paper) of the 100 ranked values of the minimum kurtosis index give different results according to the optimization method. For PSO, it seems that there is only one potential interesting projection while we visualize at least two local minima for the kurtosis and the Friedman index with Tribes. In figure 2 we visualize two interesting projections corresponding to different local optima of the kurtosis index using Tribes method. On these two plots the data are split in two parts which correspond to different regions for the oils (see the clusters defined by the nine areas below the histograms). For the same method, the Friedman index gives different projections which separate other areas. As mentioned above, PSO yields a single interesting structure by optimizing any of both indices.

As illustrated in these small examples, EPP with the Tribes algorithm is a powerful tool for discovering clusters structures if present in the data. It would be interesting to test the proposed method on higher multidimensional datasets. Once EPP has revealed the presence of clusters, the data analyst may perform some clustering algorithm in order to define precisely the clusters.

5 Conclusion

In this paper, we used two metaheuristics (PSO and Tribes) to optimize two projection indices. We showed the performance of these methods using multi-dimensional data sets for the detection of groups. The important result of our study is the performance and the efficiency of Tribes method for projection pursuit. By using several simulations, we can easily obtain several local optima of the projection index susceptible to reveal interesting structures.

The difference between PSO and Tribes is that Tribes requires no parameter to settle. The statistician has only to define the objective function and the stopping criterion. A study of this method was led by [4] and [3] who found that Tribes converges very quickly to a local optimum which is not generally the global optimum. This characteristic, which the authors [4] and [3] consider as a drawback, motivates our choice and serves perfectly our objective

Both Friedman and Kurtosis projection indices give good results on the considered examples. Concerning the computing time, we noticed that the kurtosis index is faster than the Friedman index. Although the evaluation number of the objective function is not the same for both methods (because the number of particles in the method Tribes is variable), we observed that Tribes is faster than PSO. For the small-size data sets we consider, the time is unimportant for both methods and both indices but for very large data sets the kurtosis index together with the Tribes algorithm are recommended.

References

1. Caussinus, H., Ruiz-Gazen, A.: Exploratory projection pursuit. In Data Analysis (Digital Signal and Image Processing series). Ed. G. Govaert, p. 67–92 Wiley (2009)
2. Clerc, M.: Particle swarm optimization. International Scientific and Technical Encyclopaedia Wiley, Hoboken (2006)
3. Clerc, M., Cooren, Y., Siarry, P.: Optimisation par essaim particulaire améliorée par hybridation avec un algorithme à estimation de distribution. Journées doctorales MACS. volume 5, N1 pp: 21–27 (2008)
4. Cooren, Y., Clerc, M. Siarry, P.: Performance evaluation of TRIBES, an adaptive particle swarm optimization algorithm. Swarm Intelligence 3, 149–178 (2009)
5. Cook, D., Swayne, D.F.: Interactive and Dynamic Graphics for Data Analysis. Springer Verlag. New York (2007)
6. Cook, D., Caragea, D., Honavar, H.: Visualization in classification problems. J. Antoch(Ed) Proceeding in Computational Statistics (COMPSTAT), Springer. Berlin 799–806 (2004)
7. Friedman, J. H.: Exploratory Projection Pursuit. J. Amer. Statist. Assoc. 82(1) 249–266 (1987)
8. Friedman, J. H., Tukey, J. W.: A Projection Pursuit Algorithm for Exploratory Data Analysis. IEEE Trans. Comput., C-23 881–889 (1974)
9. Huber, P.J.: Projection pursuit. The Annals of Statistics. 13(2) 435–475 (1985)
10. Jones, M.C., Sibson, R.: What is projection pursuit? (with discussion). J. Roy. Statist. Soc. A. 150, 1–36 (1987)
11. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Yuhui Shi (1995)
12. Martinez, W., Martinez, A.: Computational statistics handbook with Matlab. CRC Press (Taylor and Francis Group) (2001)
13. Nason, G.P.: Three-Dimensional Projection Pursuit. J. Roy. Statist. Soc. C. 44, 411–430 (1995)
14. Peña, D., Prieto, F.: Cluster Identification using projections. J. Amer. Statist. Assoc. 96, 456, 1433–45 (2001)
15. Sun, J.: Some Practical aspects of exploratory Projection Pursuit. SIAM J. Sci. Comput., 14, 1, 68–80 (1993)
16. Wu, J., Zhou, J., Gao, Y.: Support Vector Regression Based on Particle Swarm Optimization and Projection Pursuit Technology for Rainfall Forecasting, International Conference on Computational Intelligence and Security, vol. 1, pp.227-233 (2009)