Expert Systems With Applications 72 (2017) 49–66

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Active function Cross-Entropy Clustering

P. Spurek*, J. Tabor, K. Byrski

Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

ARTICLE INFO

Article history:
Received 22 August 2016
Revised 15 November 2016
Accepted 6 December 2016
Available online 7 December 2016

Keywords:
Clustering
Gaussian mixture models
Expectation maximization
Cross-Entropy Clustering
Active curve axis gaussian mixture model

ABSTRACT

Gaussian Mixture Models (GMM) have many applications in density estimation and data clustering. However, the models do not adapt well to curved and strongly nonlinear data, since many Gaussian components are typically needed to appropriately fit the data that lie around the nonlinear manifold.

To solve this problem we constructed the Active Function Cross-Entropy Clustering (afCEC) method, which uses Gaussians in curvilinear coordinate systems. The method has a few advantages in relation to GMM: it enables easy adaptation to clustering of complicated data sets along with a predefined family of functions and does not need external methods to determine the number of clusters, as it automatically (on-line) reduces the number of groups.

Experiments on synthetic data, Chinese characters, data from UCI repository and wind turbine monitoring systems show that the proposed nonlinear model typically obtains better results than the classical methods.

1. Introduction

Clustering plays a basic role in many parts of data engineering, pattern recognition, and image analysis. Some of the most important clustering methods are based on GMM, which in practice accommodates data with distributions that lie around affine subspaces of lower dimensions obtained by principal components (PCA) (Jolliffe, 2002), see Fig. 1(a). However, by the manifold hypothesis, real world data presented in high dimensional spaces are likely to concentrate in the vicinity of non-linear sub-manifolds of lower dimensionality (Cayton, 2005; Narayanan & Mitter, 2010). The classical approach approximates this manifold by a mixture of Gaussian distributions. Since one non-Gaussian component can be approximated by a mixture of several Gaussians (Fraley & Raftery, 1998; Śmieja & Wiercioch, 2016; Spurek & Pałka, 2016), these clusters are, in practice, represented by combination of Gaussian components. This can be seen as a form of piecewise linear approximation, see Fig. 1(a). A similar result gives the Cross Entropy Clustering (CEC) (Kamieniecki & Spurek, 2014; Tabor & Spurek, 2014; Tomczyk, Spurek, Podgórski, Misztal, & Tabor, 2016) approach, see Fig. 1(b).

In our paper we construct a general afCEC (active function Cross-Entropy Clustering) theory, which allows the clustering of data on sub-manifolds of $\mathbb{R}^d$. The motivation comes from the observation that it is often profitable to describe non-linear data by smaller numbers of components with more complicated curved shapes to obtain a better fit of data, see Fig. 1 (for more detailed analysis see Section 5). While developing this theory, we were influenced by the classical Shannon Entropy Theory (Cover & Thomas, 2012; Śmieja, 2015; Spurek & Tabor, 2013), Minimum Description Length Principle (Grünwald, 2007), Cross-Entropy Clustering (Spurek, 2017; Tabor & Spurek, 2014), Expectation Maximization (EM), Implicit Function Theorem (Krantz & Parks, 2002), and Active curve axis Gaussian Mixture Models (Ju & Liu, 2011; 2012; Zhang, Zhang, & Yi, 2005). Consequently, we present a theoretically-motivated clustering method that automatically reduces unnecessary clusters and accommodates non-linear structures. Because we have to approximate complicated structures in each step, we have to construct a numerically efficient model. Therefore, we have chosen an approach that allows for the use of an explicit formula in each step.

This paper is arranged as follows. In the next section, we present related works. Then the theoretical background of the density model will be presented (corresponding to the case of one cluster). In the fourth section, we introduce the theoretical background of the afCECmethod. We prove that the cost function decreases in every iteration, see Theorem 4.2. The last two sections we present numerical experiments.

* Corresponding author.
E-mail addresses: przemyslaw.spurek@ii.uj.edu.pl (P. Spurek), jacek.tabor@uj.edu.pl (J. Tabor), krzysztof.byrski@uj.edu.pl (K. Byrski).

http://dx.doi.org/10.1016/j.eswa.2016.12.011

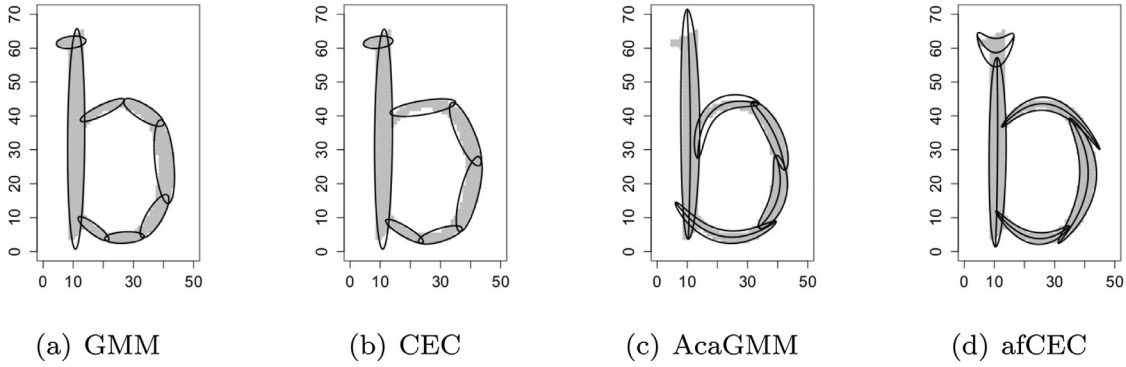


Fig. 1. Fitting a b-type set by using (a) GMM, (b) CEC, (c) AcaGMM, and (d) afCEC.

2. Related works

Density based clustering was studied by Hartigan (1975); Jain (2010); Jain and Dubes (1988); Jain, Murty, and Flynn (1999); Levin (2015); Xu and Wunsch (2009). One of the most important clustering algorithms is based on Gaussian Mixture Models (Hinton, Dayan, & Revow, 1997; Jain & Dubes, 1988; McLachlan & Krishnan, 2007; McLachlan & Peel, 2004). It is hard to overestimate the role of GMM in computer science (Hinton et al., 1997; Jain & Dubes, 1988; McLachlan & Krishnan, 2007; McLachlan & Peel, 2004); it includes object detection (Campbell, Fraley, Murtagh, & Raftery, 1997; Dasgupta & Raftery, 1998; Figueiredo & Jain, 2002; Huang, 1998; Kumar & Hebert, 2003; Samuelsson, 2004), object tracking (McKenna, Raja, & Gong, 1999; Xiong, Chen, Wang, & Huang, 2002), learning and modeling (Moghaddam & Pentland, 1997; Samuelsson, 2004), feature selection (Law, Figueiredo, & Jain, 2004; Valente & Wellekens, 2004), classification (Krawczyk, Woźniak, & Cyganek, 2014; Mukherjee et al., 1998; Povinelli, Johnson, Lindgren, & Ye, 2004), and statistical background subtraction (Basu, Naphade, & Smith, 2002; Hayman & Eklundh, 2003; Stauffer & Grimson, 1999).

2.1. Study of nonlinear data

Most of algorithms based on Gaussian distribution can be interpreted as a linear approximation of data (see Fig. 1). Since typically data in practice lies around curved structures (manifold hypotheses), algorithms which can approximate curves or manifolds are important. Principal curves and principal surfaces (Hastie & Stuetzle, 1989; Kegl, 1999; LeBlanc & Tibshirani, 1994) have been defined as self-consistent smooth curves (or surfaces in $\mathbb{R}^2$) which pass through the middle of a d-dimensional probability distribution or data cloud. They give a summary of the data and also serve as an efficient feature extraction tool. Principal curves/surfaces algorithms are typically capable of expressing a single complex manifold. If the data lies on complicated manifolds in higher dimensional spaces, these algorithms cannot fit a single manifold without special initialization.

Another method that attempts to solve the problem of fitting nonlinear manifolds is that of self-organizing maps (SOM) (Kohonen, 2001) or self-organizing feature maps (SOFM) (Kohonen, 1989). These methods are types of artificial neural networks which are trained using unsupervised learning to produce a low-dimensional (typically two-dimensional) discretized representation of the input space of the training samples, called a map. Self-organizing maps are different from other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

Kernel methods provide a powerful way of capturing non-linear relations. One of the most common, kernel PCA (KCPA) (Schölkopf, Smola, & Müller, 1998), is a non-linear version of principal component analysis (PCA) (Jolliffe, 2002) that gives an explicit low dimensional space such that the data variance in the feature space is preserved as much as possible.

The above approaches focus on finding only a single complex manifold. In general, they do not focus on the clustering method; furthermore, it is difficult to use them for dealing with clustering problems.

Kernel methods and self-organizing maps can be used as a preprocessing for classical clustering methods. In such a way spectral clustering methods was constructed (Chi & Yang, 2006; Li, Li, & Tao, 2008; Ng, Jordan, & Weiss, 2002). The classical kernel k-means (Li et al., 2008) is equivalent to KPCA prior to the conventional k-means algorithm. Most of kernel methods consist of two steps: an embedding into a feature space and a classical clustering method used on the data transformed to feature space. Therefore spectral methods are typically time consuming and use large number of parameters.

2.2. Acagmm

Zhang et al. (2005) present an adaptation of the Gaussian Mixture Model called the Active curve axis Gaussian Mixture Models (AcaGMM), which uses a nonlinear curved Gaussian probability model in clustering. Since our paper aims to solve the same task as AcaGMM, but in a greater generality, let us first explain what AcaGMM is. In its standard version, it works with data on the plane and adapts to quadratic curves. In other words, AcaGMM uses a wider class than typical Gaussians – namely Gaussians which are curved over parabolas.

AcaGMM works well in practice; however, it has major limitations. First of all, the method is not a density model.[1] Consequently, one can not use the EM procedure to minimize the AcaGMM cost function. The incorrect use of the EM method causes fundamental problems. The AcaGMM cost function does not necessarily decrease with iterations, which causes problems with the stop condition. We also do not obtain density estimation as in a correctly used EM procedure. Detailed description of AcaGMM model we present in Appendix A. It is possible to partially correct the model by taking in to consideration the Jacobian. However, even such a modification does not help because AcaGMM uses PCA and regression in each cluster.[2] Moreover, AcaGMM is

[1] The curved Gaussian function used in AcaGMM is not a density, as the Jacobian of the transformation is not taken in to consideration (for more details see Section Appendix A).

[2] By applying two methods (PCA and regression) separately, we do not minimize either of them. Consequently, even if we apply densities in AcaGMM, the cost function will not decrease with iterations.

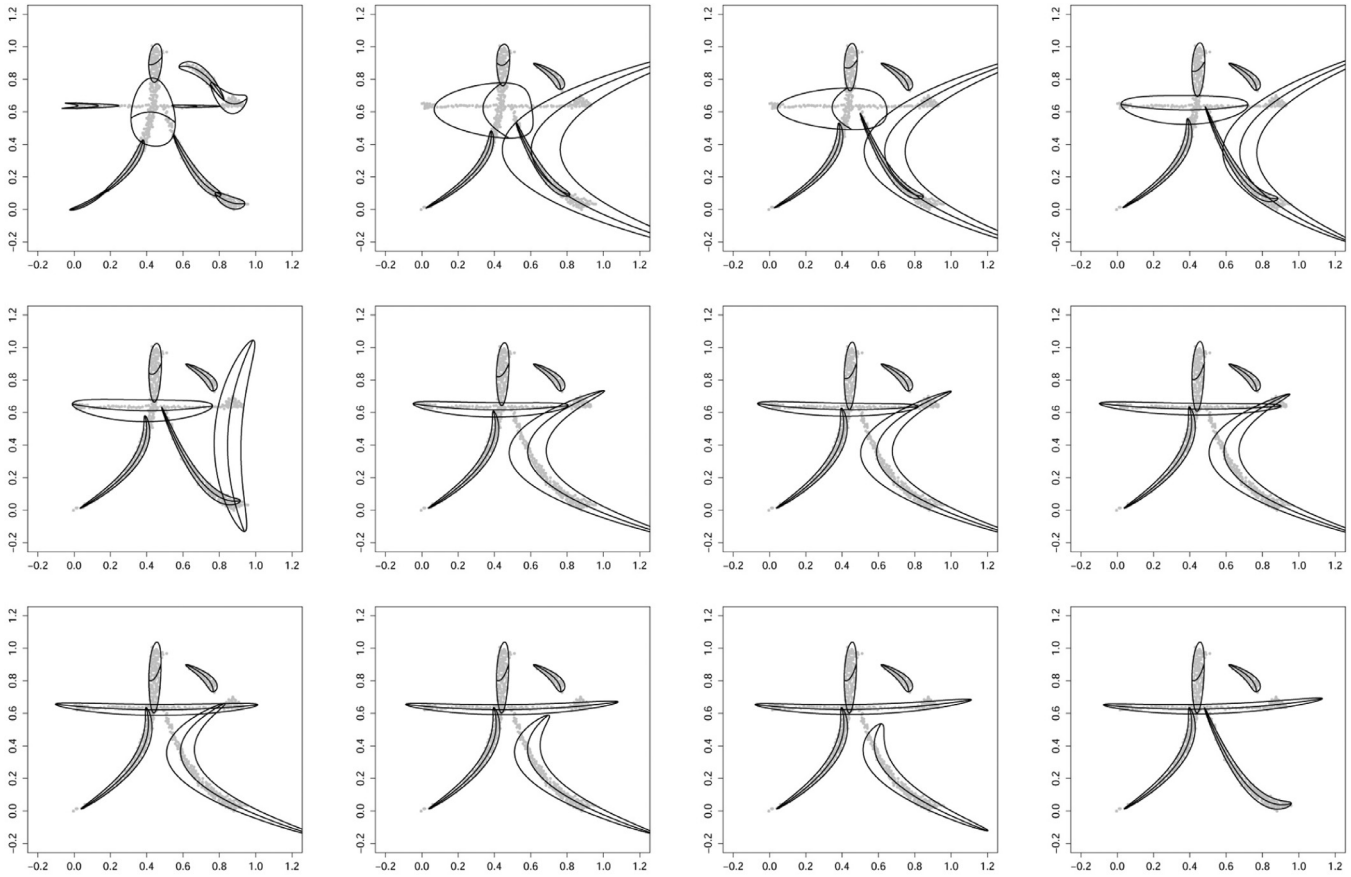


Fig. 2. The convergence process of afCECon a Chinese character with initial $k = 10$, which is reduced to $k = 5$.

naturally restricted to quadratic functions.[3] Furthermore, the use of the method in dimensions higher then two, although theoretically possible, is impractical from a numerical point of view.

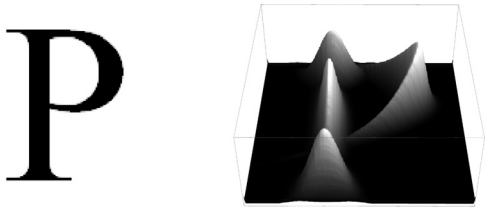
In this paper, we propose the Active Function Cross-Entropy Clustering (afCEC) method (see Fig. 1(d)) which is based on the Cross-Entropy Clustering (CEC) model.[4] Thanks to the simplicity of the density model, we are able to construct an algorithm which is easy to adapt to the case with a higher dimension. Moreover, the afCECmethod is able to reduce unnecessary clusters. In Fig. 2, we present the convergence process of afCECwith the initial number of clusters at $k = 10$, which is reduced to $k = 5$. The afCEC algorithm uses densities, and therefore we can interpret it as a density estimation, see Fig. 3. Consequently, we can compare it with other density based algorithms like GMM or CEC by using the log-likelihood function. Experiments on synthetic data, Chinese characters, data from the UCI repository and from wind turbine monitoring systems (see Section 5) show that afCEC better describes the intricate structure of data by using fewer parameters.

3. Theory: adapted gaussians

In this section, we focus on f-adapted Gaussian distributions, where $f \in \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$ is a continuous function. The goal of this approach is to transform a normal distribution (which assumes the intrinsic linearity of the model) to the case of manifolds given by the graph of the function f. The above model will be used in the afCECmethod as a representation of each cluster.

[3] This is caused by the fact that the projection onto the graph and arc length of the curve must be computed.

[4] Instead of the Expectation Maximization (EM).



(a) The image of P letter. (b) An density estimation.

Fig. 3. Density estimation obtained by afCEC.

3.1. Toy example in $\mathbb{R}^2$

We begin with an illustration of our idea in the two-dimensional case. Let us recall that a two-dimensional Gaussian density with mean $\mathrm{m} = [m_1, m_2]^T$ and covariance matrix $\Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}$ is given by the following formula:

$$N(\mathrm{m}, \Sigma)(\mathrm{x}) = N(m_1, \sigma_1^2)(x_1) \cdot N(m_2, \sigma_2^2)(x_2), \tag{1}$$

where in the one-dimensional case we have:

$$N(m, \sigma^2)(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{|x-m|^2}{2\sigma^2}\right).$$

In the case of a "curvilinear" coordinate system, we adapt the Gaussian density to the arbitrary given function $f \in \mathcal{C}(\mathbb{R}, \mathbb{R})$. For each point, we use the Euclidean distance along the second coordinate to curve f instead of applying the coordinates on the

canonical basis

$$N(\mathrm{m}, \Sigma, f)([x_1, x_2]^T) = N(m_1, \sigma_1^2)(x_1) \cdot N(m_2, \sigma_2^2)(x_2 - f(x_1)), \tag{2}$$

see Fig. 5.

Although at first the difference between formulas (2) and (1) seems small, the above modification allows us to describe nonlinear data. Level sets of curved Gaussian distributions in the case of different functions in $\mathbb{R}^2$ are presented in Fig. 5.

For the convenience of the reader, and to present the basic difference between afCEC and AcaGMM we will briefly describe the AcaGMM model. Since AcaGMM works in the two-dimensional case (in cases with higher dimensions, the authors use PCA to reduce problems to 2D) with parabolas ($f(x) = ax^2 + b$ for $a, b \in \mathbb{R}$), we will restrict our discussion to this situation. Let $f(x) = ax^2 + b$ for $a, b \in \mathbb{R}$ be given and let $\mathrm{x} = [x_1, x_2]^T \in \mathbb{R}^2$. The AcaGMM approach uses the orthogonal projection of the point x onto the parabola f, which is denoted by $p_f(\mathrm{x})$, and the arc length between $p_f(\mathrm{x})$ and m, which is denoted by $l_f(p_f(\mathrm{x}), \mathrm{m})$. Consequently, the AcaGMM function is given by

$$N(\mathrm{m}, \Sigma, f)(\mathrm{x}) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(\frac{-l_f(p_f(\mathrm{x}), \mathrm{m})^2}{2\sigma_1^2}\right) \frac{1}{\sqrt{2\pi}\sigma_2} \exp \times\left(\frac{-\|p_f(\mathrm{x}) - \mathrm{x}\|^2}{2\sigma_2^2}\right). \tag{3}$$

Although this approach is very intuitive, it causes some basic problems. It is very hard (or even impossible) to give explicit formulas for orthogonal projections and arc lengths for more complicated curves in higher dimensional spaces. Thus calculations are complicated (from a numerical point of view) and, consequently, possible generalizations of AcaGMM are limited. Moreover, the function which was used in AcaGMM, see formula (3), is not a density, as the Jacobian of the respective transformation was not included.

The practical difference between AcaGMM and our approach in $\mathbb{R}^2$ in the case of one cluster described by a parabola is rather small[5], see Fig. 4. Nevertheless, our model is more flexible as we can use an arbitrary class of functions in an arbitrary dimension for which the least square methods work, see Theorem 4.2.

3.2. f-Adapted gaussian density

In this subsection, the general notion of the f-adapted Gaussian is presented. Moreover, we show (see Theorem 3.1) that under weak assumptions, the class of adapted Gaussians contains a class of classical normal distributions.

Let us recall that the standard Gaussian density in $\mathbb{R}^d$ is defined by

$$N(\mathrm{m}, \Sigma)(\mathrm{x}) = \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}\|\mathrm{x} - \mathrm{m}\|_\Sigma^2\right),$$

where m denotes the mean, Σ is the covariance matrix, and $\|\mathrm{v}\|_\Sigma^2 = \mathrm{v}^T \Sigma^{-1} \mathrm{v}$ is the square of the Mahalanobis norm.

In our work, we use a multidimensional Gaussian density in a curvilinear coordinate system which is spread along the function $f : \mathbb{R}^{d-1} \to \mathbb{R}$ (f-adapted Gaussian density). We treat one of the variables separately. In such a case we consider only those $\Sigma \in \mathcal{M}_d(\mathbb{R})$ (where $\mathcal{M}_d(\mathbb{R})$ denotes the set of d-dimensional square, symmetrical, and positive define matrices) which have the diagonal block matrix form

$$\Sigma = \begin{bmatrix} \Sigma_{\hat{l}} & 0 \\ 0 & \Sigma_l \end{bmatrix},$$

where $\Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R})$ and $\Sigma_l > 0$. For $\mathrm{x} = [x_1, \ldots, x_d]^T \in \mathbb{R}^d$ and $l \in \{1, \ldots, d\}$, we will use the notation

$$\mathrm{x}_{\hat{l}} = [x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_d]^T \in \mathbb{R}^{d-1}.$$

Now, we will give the mathematically formal definition of the f-adapted Gaussian function.

Definition 3.1. Let $f \in \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$, $\Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R})$, $\Sigma_l > 0$, $\mathrm{m} = [\mathrm{m}_{\hat{l}}, m_l]^T \in \mathbb{R}^d$ be given. The f-adapted Gaussian density for $\Sigma_{\hat{l}}$, Σ_l, $l \in \{1, \ldots, d\}$ and m is defined as follows

$$N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f)(\mathrm{x}) = N(\mathrm{m}_{\hat{l}}, \Sigma_{\hat{l}})(\mathrm{x}_{\hat{l}}) \cdot N(m_l, \Sigma_l)(x_l - f(\mathrm{x}_{\hat{l}})) \tag{4}$$

Level sets for f-adapted Gaussian distributions with respect to different types of functions are presented in Fig. 5. The f-adapted Gaussian function is a density model, see Observation 3.1, which has profound consequences for the convergence of the minimization procedure.

Observation 3.1. The f-adapted Gaussian function $N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f)(\mathrm{x})$, where $f \in \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$, $\Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R})$, $\Sigma_l \in \mathbb{R}$, $\mathrm{m} \in \mathbb{R}^d$ is a density.

Proof. Assume, without a loss of generality, that $l = d$. Let $N(\mathrm{m}, \Sigma)$ be a d-dimensional Gaussian density such, that $\Sigma = \begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ 0 & \Sigma_d \end{bmatrix}$, where $\Sigma_{\hat{d}} \in \mathcal{M}_{d-1}(\mathbb{R})$, $\Sigma_d > 0$, $\mathrm{m} \in \mathbb{R}^d$. We have to show that

$$\int_{\mathbb{R}^d} N(\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f)(\mathrm{x}) d\mathrm{x} = 1.$$

Let us consider a transformation

$$[y_1, \ldots, y_d]^T = [x_1, \ldots, x_{d-1}, x_d - f(\mathrm{x}_{\hat{d}})]^T,$$

which Jacobian equals

$$J(x_1, \ldots, x_d) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ \dfrac{\partial f(\mathrm{x}_{\hat{d}})}{\partial x_1} & \dfrac{\partial f(\mathrm{x}_{\hat{d}})}{\partial x_2} & \cdots & \dfrac{\partial f(\mathrm{x}_{\hat{d}})}{\partial x_{d-1}} & 1 \end{bmatrix}.$$

It is easy to show that $\det(J(\mathrm{x})) = 1$. In such a case we have

$$\int_{\mathbb{R}^d} N(\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f)(\mathrm{x}) d\mathrm{x} = \int_{\mathbb{R}^d} N(\mathrm{m}, \Sigma)(\mathrm{x}) \det(J(\mathrm{x})) d\mathrm{x} = \int_{\mathbb{R}^d} N(\mathrm{m}, \Sigma)(\mathrm{x}) d\mathrm{x} = 1.$$

Consequently, $N(\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f)$ is a density. □

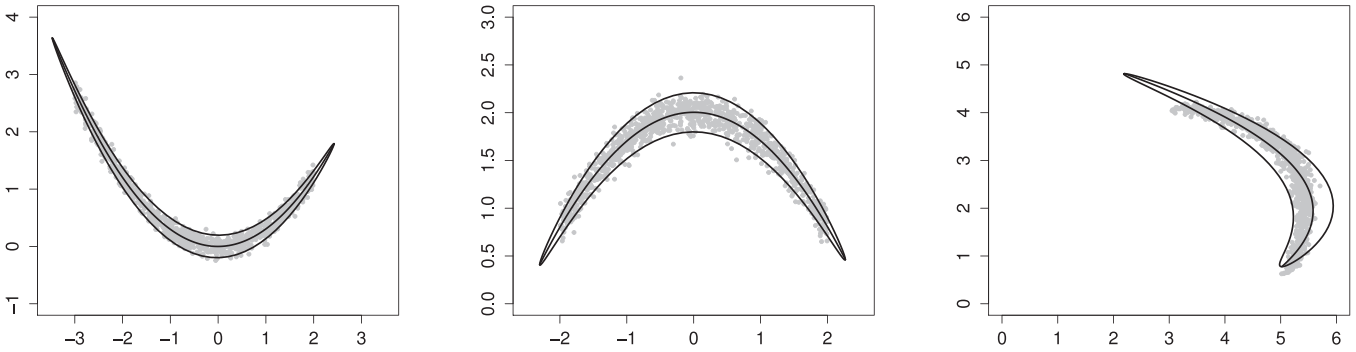
In the basic form of the CEC algorithm, we are looking for the optimal Gaussian function in the family of all d-dimensional Gaussian densities $\mathcal{G}(\mathbb{R}^d)$. In the case of afCEC, we describe each cluster by the f-adapted Gaussian function. Consequently, we need to find optimal density in the class of all curved Gaussians. For the given $f : \mathbb{R}^{d-1} \to \mathbb{R}$, we denote the *family of all f-adapted Gaussian functions* by

$$\mathcal{G}_l[f] = \left\{N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f) : \mathrm{m} \in \mathbb{R}^d, \Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R}), \Sigma_l > 0\right\}.$$

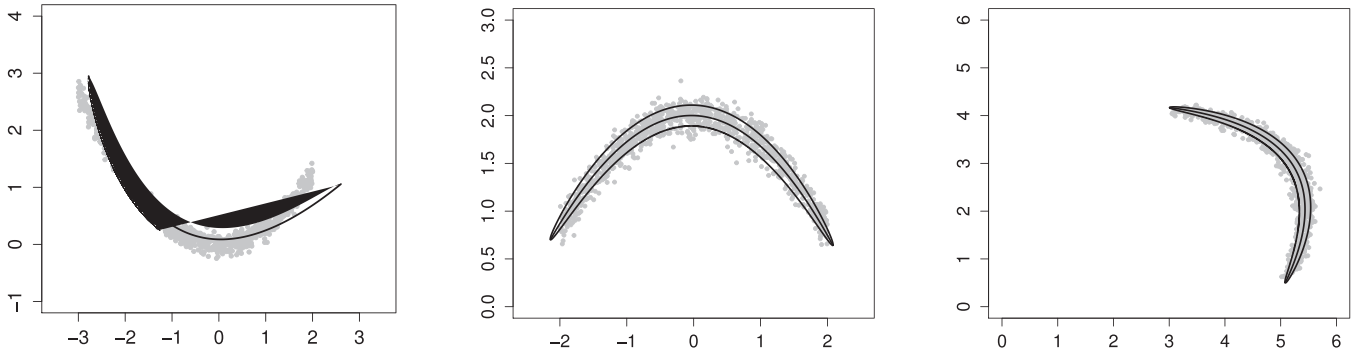
In the afCEC algorithm, we describe clusters by generalized Gaussian distributions from $\mathcal{G}_l[f]$ where f is in some class of functions (we can use any class of functions for which the regression procedure works) and $l \in \{1, \ldots, d\}$. Therefore, we will need one more definition. For the family $\mathcal{F} \subset \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$, we define

$$\mathcal{G}_l[\mathcal{F}] = \bigcup_{f \in \mathcal{F}} \mathcal{G}_l[f].$$

[5] In our case, we use the parabola $ax^2 + bx + c$ instead of $ax^2 + c$ since our method does not apply the change of coordinates given by PCA.

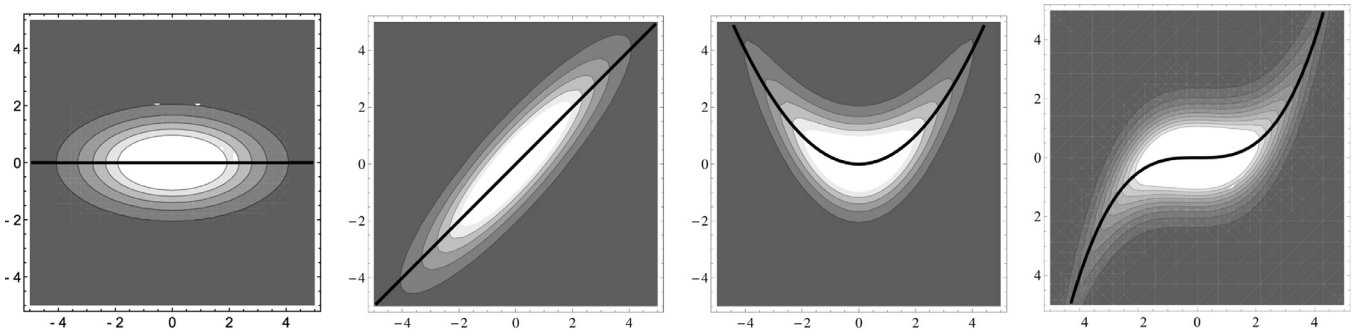


(a) The afCEC method.



(b) The AcaGMM method.

Fig. 4. Representation of generalized Gaussian distributions of AcaGMM and afCEC in the case of $\mathbb{R}^2$ and parabolas.



(a) $f(x) = 0$ (b) $f(x) = x$ (c) $f(x) = \frac{1}{8}x^2$ (d) $f(x) = \frac{1}{16}x^3$

Fig. 5. Level sets for f-adapted Gaussian distribution.

If $\mathcal{F}$ contains linear functions, then the family coincides with the class of all classical Gaussian distributions.

Theorem 3.1. *Let* $\mathcal{F} = \{f : \mathbb{R}^{d-1} \to \mathbb{R} : f(\mathrm{x}) = \mathrm{v}^T \cdot \mathrm{x} \text{ for } \mathrm{v} \in \mathbb{R}^{d-1}\}$ *be the family of all linear transformations from* $\mathbb{R}^{d-1}$ *into* $\mathbb{R}$. *Then,*

$\mathcal{G}_l[\mathcal{F}] = \mathcal{G}(\mathbb{R}^d)$ for all $l \in \{1, \ldots, d\}$.

Proof. See Appendix. □

This implies that afCECis a natural extension of the standard CEC algorithm, because for $\mathcal{F}$ containing only linear functionals, we obtain exactly the standard Gaussian densities. On the other hand, for wider classes of functions $\mathcal{F}$ we can detect curved clusters, which describe groups concentrated around manifolds which are not necessarily linear.

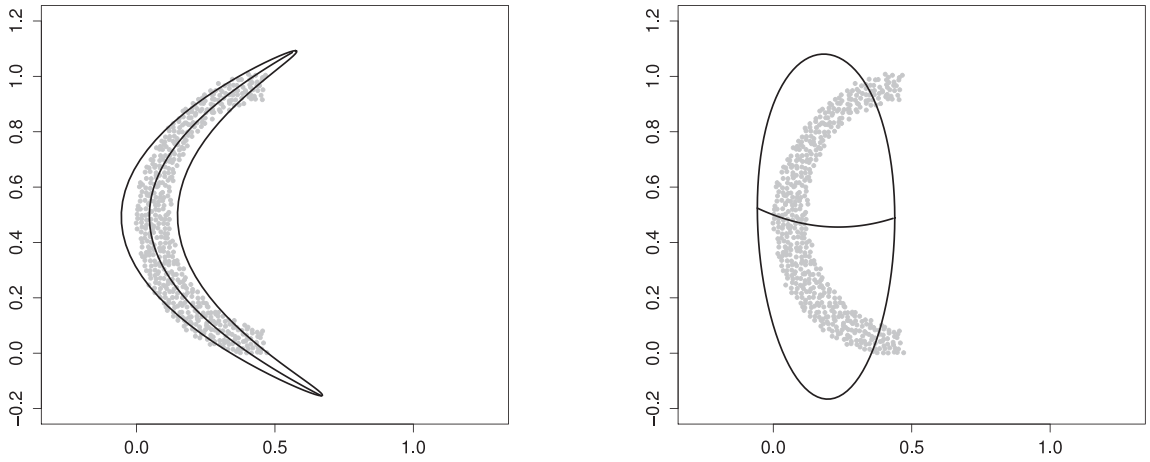
The following observation is a direct corollary from Theorem 3.1.

Corollary 3.1. *Let* $\mathcal{F} \subset \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$ *contain the family of all linear functionals from* $\mathbb{R}^{d-1}$ *into* $\mathbb{R}$. *Then*

$\mathcal{G}(\mathbb{R}^d) \subset \mathcal{G}_l[\mathcal{F}]$,

for $l \in \{1, \ldots, d\}$.

In the previous considerations, we assumed that one variable was chosen to be dependent. Since, in the case of the $\mathcal{F}$-adaptive Gaussian density, all computations are applied in the canonical basis, we can verify all possible dependent variable choices. Our idea for checking all coordinates in the canonical basis came from the Implicit Function Theorem (Krantz & Parks, 2002).



(a) The c-type set and parabola fitted with assumption that x is the dependent variable.

(b) The c-type set and parabola fitted with assumption that y is the dependent variable.

Fig. 6. Visualization of optimal $\mathcal{F}$-adaptive Gaussian density with respect to two different choices of dependent variables in the case of the family $\mathcal{F}$ containing parabolas.

Observation 3.2. Under reasonable assumptions, for a manifold $M = \{x : F(x) = 0\}$ where $F : \mathbb{R}^d \to \mathbb{R}$ and a point $\bar{x} \in M$, we can find $l \in \{1, \ldots, d\}$ and $f : \mathbb{R}^{d-1} \to \mathbb{R}$ such that locally in the neighborhood U of $\bar{x}$

$$U \cap M = \{x \in U : F(x) = 0\} = \{(x_1, \ldots, x_{l-1}, f(x_{\hat{l}}), x_{l+1}, \ldots, x_d) \text{ for } (x_1, \ldots, x_{l-1}, x_{l+1}, \ldots, x_d) \text{ close to } \bar{x}_{\hat{l}}\}.$$

Consequently, for data $X \subset \mathbb{R}^d$, we search for $l \in \{1, \ldots, d\}$ and f such that X can be optimally approximated by the set

$$(x_1, \ldots, x_{l-1}, f(x_{\hat{l}}), x_{l+1}, \ldots, x_d) \text{ for } x \in X.$$

In such a situation, an lth coordinate is chosen as a dependent variable and the rest of them become explanatory variables.

The above observation explains the intuition connected with checking all coordinates in the canonical basis instead of finding a local coordinate system. Let us now present an example of such a procedure.

Example 3.1. Let us consider a c-type set, see Fig. 6. When using the canonical basis of $\mathbb{R}^2$, we have to consider two possible estimated curves (in our case, parabolas). We can treat x as a dependent variable, see Fig. 6(a), or we can choose y as a dependent variable, see Fig. 6(b). If we assume that the dependent variable is x, we obtain the parabola $x = 1.4755y^2 - 1.4602y + 0.4078$, and the sum of the squared errors is equal to 1.420948. On the other hand, if y is the dependent coordinate, we have $y = 0.8x^2 - 0.3756x + 0.4997$ with the squared errors equal to 53.35997. Consequently, the optimal coordinate system is given by using x as the dependent variable. Moreover, it is easy to see (compare Fig. 6(a) and (a)) that in the case of choosing x as a dependent variable, we obtain a better fitting of data.

In the above example, we consider only $\mathbb{R}^2$, but for $X \subset \mathbb{R}^d$ we have to consider d different possible choices of dependent variables.

For the family $\mathcal{F} \subset \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$, we define the family of $\mathcal{F}$-adapted Gaussian distributions with all the possible choices of dependent variables by

$$\mathcal{G}[\mathcal{F}] = \bigcup_{l=1}^{d} \mathcal{G}_l[\mathcal{F}].$$

4. Theoretical background of afCEC

In this section, the theoretical background of afCECwill be presented. First, we introduce the cost function that will be minimized by the algorithm. Then, we prove that the optimal function describing each cluster can be obtained by the least square regression (Björck, 1996). We will end by describing the full algorithm of afCEC.

Our method is based on the CEC approach. Therefore, we start with a short introduction to the method (for a more detailed explanation we refer the reader to Tabor & Spurek (2014)). Since CEC is similar to EM in many aspects, let us first recall that, in general, EM aims to find $p_1, \ldots, p_k \geq 0$, $\sum_{i=1}^{k} p_i = 1$ and $f_1, \ldots, f_k$ Gaussian densities (where k is given beforehand and denotes the number of densities for which the convex combination builds the desired density model) such that the convex combination

$$f = p_1 f_1 + \cdots + p_k f_k$$

optimally approximates the scatter of our data X with respect to the MLE cost function

$$\mathrm{MLE}(f, X) = -\sum_{x \in X} \ln(p_1 f_1(x) + \cdots + p_k f_k(x)). \tag{5}$$

The EM procedure consists of the Expectation and Maximization steps. While the Expectation step is relatively simple, the Maximization step usually needs complicated numerical optimization even for relatively simple Gaussian models (Banfield & Raftery, 1993; Celeux & Govaert, 1995; Davis-Stober, Broomell, & Lorenz, 2007). A goal of CEC is to minimize the cost function, which is a minor modification of that given in (5) by substituting the sum with the maximum:

$$\mathrm{CEC}(f, X) = -\sum_{x \in X} \ln(\max(p_1 f_1(x), \ldots, p_k f_k(x))). \tag{6}$$

Instead of focusing on the density estimation as its main task, CEC aims itself directly at the clustering problem. It occurs that at the small cost of having a minimally worse density approximation (Tabor & Spurek, 2014), we gain speed in implementation[6] and the ease of using more complicated density models. Roughly speaking,

[6] We can often use the Hartigan approach to clustering, which is faster and typically finds better minima.

this is more advantageous because the models do not mix with each other since we take the maximum instead of the sum.

To apply CEC, we need to introduce the cost function which we want to minimize. In the case of splitting $X \subset \mathbb{R}^d$ into $X_1, \ldots, X_k$ so that we code elements of X_i using a function from the family of all Gaussian densities $\mathcal{G}(\mathbb{R}^d)$, the mean code-length of a randomly chosen element x equals

$$E(X_1, \ldots, X_k; \mathcal{G}(\mathbb{R}^d)) = \sum_{i=1}^{k} p_i \cdot \left(-\ln(p_i) + H^{\times}(X_i \| \mathcal{G}(\mathbb{R}^d))\right) \tag{7}$$

where $p_i = \frac{|X_i|}{|X|}$. The formula uses the Cross-Entropy of a data set with respect to the family $\mathcal{G}(\mathbb{R}^d)$.

The aim of CEC is to split of dataset X into sets X_i which minimize the function given in (7). Our goal is to calculate an explicit formula for the cost function in the case of f-adapted Gaussian densities.

4.1. Cost function of one cluster

In this subsection we will focus on the situation of one cluster X. In such a case, we usually understand the data as a realization of a random variable, where we use

$$\text{mean}(X) = \frac{1}{n}\sum_{\mathrm{x}\in X} \mathrm{x},$$

$$\text{cov}(X) = \frac{1}{n}\sum_{\mathrm{x}\in X} (\mathrm{x} - \text{mean}(X))(\mathrm{x} - \text{mean}(X))^T,$$

as an estimator for the mean and covariance. For $X \subset \mathbb{R}^d$, we denote $X_{\hat{l}} = \{\mathrm{x}_{\hat{l}} : \mathrm{x} \in X\}$, the set containing vectors from X with removed l coordinate, and $X_l = \{x_l : [x_1, \ldots, x_d]^T \in X\}$. For the function $f : \mathbb{R}^{d-1} \to \mathbb{R}$, we put

$$X_l^f = \{x_l - f(\mathrm{x}_{\hat{l}}) : \mathrm{x} \in X\} \subset \mathbb{R}.$$

As was previously mentioned, CEC uses the Cross-Entropy of data set X with respect to the Gaussian family $\mathcal{G}(\mathbb{R}^d)$.

Theorem 4.1. *Let $X \subset \mathbb{R}^d$ be given. Then,*

$$\begin{aligned} H^{\times}(X\|\mathcal{G}(\mathbb{R}^d)) &= \inf\{H^{\times}(X\|g) : g \in \mathcal{G}(\mathbb{R}^d)\} \\ &= \frac{d}{2}\ln(2\pi e) + \frac{1}{2}\ln(\det(\text{cov}(X))). \end{aligned}$$

The CEC algorithm will be used for a family of $\mathcal{F}$-adapted Gaussian densities. In such a case, the cost function is described by the following theorem.

Theorem 4.2. *Let $X \subset \mathbb{R}^d$ and a function $f \in \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$ be given. Then,*

$$\begin{aligned} H^{\times}(X\|\mathcal{G}_l[f]) = &\frac{d}{2}\ln(2\pi e) + \frac{1}{2}\ln(\det(\text{cov}(X_{\hat{l}}))) \\ &+\frac{1}{2}\ln(\text{cov}(X_l^f)) \text{ for } l \in \{1, \ldots, d\}. \end{aligned}$$

Proof. Consider $N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f) \in \mathcal{G}[f]$, where $\Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R})$ is an arbitrary symmetric positive matrix, $\Sigma_l \in (0, \infty)$ and $\mathrm{m} = [\mathrm{m}_{\hat{l}}, m_l]^T \in \mathbb{R}^d$. The assertion of the proposition is a simple consequence of

$$\begin{aligned} H^{\times}(X\|N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f)) &= -\frac{1}{|X|}\sum_{\mathrm{x}\in X} \ln(N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f)(\mathrm{x})) \\ &= -\frac{1}{|X|}\sum_{\mathrm{x}\in X} \ln\left(N(\mathrm{m}_{\hat{l}}, \Sigma_{\hat{l}})(\mathrm{x}_{\hat{l}}) N(m_l, \Sigma_l)(x_l - f(\mathrm{x}_{\hat{l}}))\right) \\ &= -\frac{1}{|X|}\sum_{\mathrm{x}\in X} \ln\left(N(\mathrm{m}_{\hat{l}}, \Sigma_{\hat{l}})(\mathrm{x}_{\hat{l}})\right) \\ &\quad -\frac{1}{|X|}\sum_{\mathrm{x}\in X} \ln\left(N(m_l, \Sigma_l)(x_l - f(\mathrm{x}_{\hat{l}}))\right) \\ &= H^{\times}(X_{\hat{l}}\|N(\mathrm{m}_{\hat{l}}, \Sigma_{\hat{l}})) + H^{\times}(X_l^f\|N(m_l, \Sigma_l)). \end{aligned}$$

We can use Theorem 4.1 for both summands separately:

$$\begin{aligned} H^{\times}(X\|\mathcal{G}[f]) &= H^{\times}(X_{\hat{l}}\|\mathcal{G}(\mathbb{R}^{d-1})) + H^{\times}(X_l^f\|\mathcal{G}(\mathbb{R})) \\ &= \frac{d-1}{2}\ln(2\pi e) + \frac{1}{2}\ln\left(\det\left(\text{cov}(X_{\hat{l}})\right)\right) + \frac{1}{2}\ln(2\pi e) \\ &\quad +\frac{1}{2}\ln\left(\frac{1}{|X|}\sum_{\mathrm{x}\in X}(x_l - f(\mathrm{x}_{\hat{l}}) - m_l)^2\right) \\ &= \frac{l}{2}\ln(2\pi e) + \frac{1}{2}\ln(\det(\text{cov}(X_{\hat{l}}))) + \frac{1}{2}\ln(\text{cov}(X_l^f)). \quad \square \end{aligned}$$

As a corollary from the above theorem, we obtain the optimal from the Cross-Entropy point of view function for $l \in \{1, \ldots, d\}$, which describes a cluster that can be obtained by the least squares method (Björck, 1996).

Observation 4.1. Let $X \subset \mathbb{R}^d$ be a data set and the family of functions $\mathcal{F} \subset \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$ be given. Then

$$\begin{aligned} &\text{argmin}\{f \in \mathcal{F} : H^{\times}(X\|\mathcal{G}_l[f])\} \\ &= \text{argmin}\left\{f \in \mathcal{F} : \sum_{\mathrm{x}\in X} |x_l - f(\mathrm{x}_{\hat{l}}) - m_l|^2\right\}, \end{aligned}$$

where $m_l = \text{mean}(X_l)$ and $l \in \{1, \ldots, d\}$.

Consequently, we minimize cross-entropy by finding a least squares estimation. Moreover, if $\mathcal{F}$ is a set of functions which are invariant under the operations $f \to a + f$ for any a, it is enough to find $\text{argmin}_{f\in\mathcal{F}} |x_l - f(\mathrm{x}_{\hat{l}})|^2$.

Corollary 4.1. *Let $X \subset \mathbb{R}^d$ be a data set, and let a family of functions $\mathcal{F} \subset \mathcal{C}(\mathbb{R}^{d-1}, \mathbb{R})$ be invariant under the operations $f \to a + f$ for $a \in \mathbb{R}$. Let $\bar{f}_l \in \mathcal{F}$ for $l \in \{1, \ldots, d\}$ be such that $\bar{f}_l = \text{argmin}\{f \in \mathcal{F} : |x_l - f(\mathrm{x}_{\hat{l}})|^2\}$. Then,*

$$\begin{aligned} &\min_{f\in\mathcal{F}} H^{\times}(X\|\mathcal{G}_l[f]) \\ &= \frac{d}{2}\ln(2\pi e) + \frac{1}{2}\ln(\det(\Sigma_{\hat{l}})) + \frac{1}{2}\ln\left(\frac{1}{n}\sum_{\mathrm{x}\in X} |x_l - \bar{f}(\mathrm{x}_{\hat{l}})|^2\right), \end{aligned}$$

where $\Sigma_{\hat{l}} = \text{cov}(X_{\hat{l}})$ and $l \in \{1, \ldots, d\}$.

The above theorem guarantees that the cost function is decreasing during iterations when we apply the regression procedure for extracting a function which describes clusters. The analogue of this result does not hold for AcaGMM (PCA is used for finding a local coordinate system). Consequently, in afCEC(contrary to AcaGMM), we are able to construct a simple stop condition.

When we have all of the possible dependent variables we need to find the optimal one. More precisely, we find parameters which minimize Cross-Entropy respectively to family $\mathcal{G}[\mathcal{F}]$

$$H^{\times}(X\|\mathcal{G}[\mathcal{F}]) = \min_{l\in\{1,\ldots,d\}} H^{\times}(X\|\mathcal{G}_l[\mathcal{F}]).$$

In conclusion, for one cluster $X \subset \mathbb{R}^d$, we estimate the parameters of the model in two steps. First, we consider all of the possible choices of dependent variables and calculate functions f_l (corresponding with relations $x_l = f_l(\mathrm{x}_{\hat{l}})$), means $m_l = \text{mean}(X_l^{f_l})$, $\mathrm{m}_{\hat{l}} = \text{mean}(X_{\hat{l}})$ and covariances $\Sigma_{\hat{l}} = \text{cov}(X_{\hat{l}})$, $\Sigma_l = \text{cov}(X_l^{f_l})$ for $l \in \{1, \ldots, d\}$. More precisely, we find f_l-adapted Gaussian distributions

$$N([\mathrm{m}_{\hat{l}}, 0]^T, \Sigma_{\hat{l}}, \Sigma_l, f_l),$$

which realize a minimum of cross-entropy

$$H^{\times}(X\|\mathcal{G}_l[\mathcal{F}]),$$

for $l \in \{1,\dots,d\}$. Then we determine the optimal dependent variable

$$j = \underset{l\in\{1,\dots,d\}}{\operatorname{argmin}} \left\{H^{\times}(X\|\mathcal{G}_l[\mathcal{F}])\right\}.$$

Consequently, our data set is represented by the function, mean, and covariance matrix

$$f = f_j, \qquad \mathrm{m} = [\mathrm{m}_{\hat{j}}, 0], \qquad \Sigma = \begin{bmatrix} \Sigma_{\hat{j}} & 0 \\ 0 & \Sigma_j \end{bmatrix},$$

where subscript $j \in \{1,\dots,d\}$ denotes the dependent variable in the cluster. The above parameters minimize the cost function of one cluster $H^{\times}(X\|\mathcal{G}[\mathcal{F}])$.

4.2. Optimization problem and afcec algorithm

In the previous subsections we presented the process of describing one cluster by optima parameters: means, covariances, and regression functions by verifying which coordinates should be dependent. Now, we are ready to introduce the afCEC optimization problem.

Optimization Problem 4.1. Divide the data set $X \subset \mathbb{R}^d$ into k pairwise disjoint groups $X_1,\dots,X_k$ $(X = X_1 \cup \dots \cup X_k)$ such that cost function

$$E(X_1,\dots,X_k;\mathcal{G}[\mathcal{F}]) = \sum_{i=1}^{k} p_i\big(-\ln(p_i) + H^{\times}(X_i\|\mathcal{G}[\mathcal{F}])\big), \tag{8}$$

where $p_i = \frac{|X_i|}{|X|}$, is minimal.

Since our cost function is given by cross-entropy, it is bounded from below by the Shannon entropy (Cover, Thomas, & Wiley, 1991). Moreover, typically to most clustering problem it has many local minima's. Our goal is to find the optimal one, which value is close to the total minimum of the cross-entropy function. It is NP-hard problem (Mahajan, Nimbhorkar, & Varadarajan, 2009) to find the optimal clustering for k-means even in the scalar case. Since our method generalizes CEC, and CEC in the limiting case reduces to k-means Tabor and Spurek (2014), it is also an NP-hard problem to find the optimal solution. Thus, similarly to classical k-means, result of afCECstrongly depends on the initialization. To avoid poor initializations, we use a k-means++ approach, which was introduced by Arthur and Vassilvitskii (2007). We also start our algorithm at least ten times and choose the optimal solution.

Lloyd's method uses two steps which are applied simultaneously. In the first one, we estimate the parameters of means, covariances, and regression functions. It is important to verify all of the choices of dependent variables and use $j = 1,\dots,d$, which minimizes Cross-Entropy

$$\underset{l=1,\dots,d}{\operatorname{argmin}} H^{\times}(X_i\|\mathcal{G}_l[\mathcal{F}]).$$

In the second step, we construct a new division of X by adding the points to the closest cluster, or, more precisely, to the closest curved Gaussian density. Consequently, we assign the point $\mathrm{x} \in X$ to the cluster $i \in \{1,\dots,k\}$ such that

$$-\ln(p_i) - \ln\big(N([\mathrm{m}_{\hat{i}}, 0]^T, \Sigma_{\hat{i}}, \Sigma_i, f_i)(\mathrm{x})\big), \tag{9}$$

is minimal. We apply the above steps simultaneously until the change of cost function (8) is small, see Algorithm 1.

We compared the computational times between afCECand alternative methods: CEC implemented in R package **CEC** (Kamieniecki & Spurek, 2014; Spurek, Kamieniecki, Tabor, Misztal, Śmieja, 2016)

Algorithm 1 afCEC.

Input
number of clusters $k > 0$
curve family $\mathcal{F}$
stop condition $\varepsilon > 0$
dataset X (d - dimension of data)
initial conditions
obtain initial clustering $X_1,\dots,X_k$
obtain probabilities $p_i = \frac{|X_i|}{|X|}$ for $i \in \{1,\dots,k\}$
obtain parameters in each cluster f_i, mean $\mathrm{m}_{\hat{i}}$ and covariances $\Sigma_{\hat{i}}$, Σ_i (choosing the best orientation)
obtain cost function
$h_0 = \sum_{i=1}^{k} p_i(-\ln(p_i) + H^{\times}(X_i\|N([\mathrm{m}_{\hat{i}}, 0]^T, \Sigma_{\hat{i}}, \Sigma_i, f_i)))$
repeat
$n = 0$
obtain new clustering $X_1,\dots,X_k$ by matching elements to the cluster such that $-\ln(p_i) - \ln(N([\mathrm{m}_{\hat{i}}, 0]^T, \Sigma_{\hat{i}}, \Sigma_i, f_i))$ is minimal
delete unnecessary clusters ($|X_i| < 1\% \cdot |X|$) by adding elements to the closest existing one
update parameter k
$n = n + 1$
obtain new probabilities $p_i = \frac{|X_i|}{|X|}$ for $i \in \{1,\dots,k\}$
obtain new parameters of each cluster f_i, mean $\mathrm{m}_{\hat{i}}$ and covariances $\Sigma_{\hat{i}}$, Σ_i (choosing the best orientation)
obtain new cost function
$h_n = \sum_{i=1}^{k} p_i(-\ln(p_i) + H^{\times}(X_i\|N([\mathrm{m}_{\hat{i}}, 0]^T \Sigma_{\hat{i}}, \Sigma_i, f_i)))$
until $h_n \geq h_{n-1} - \varepsilon$

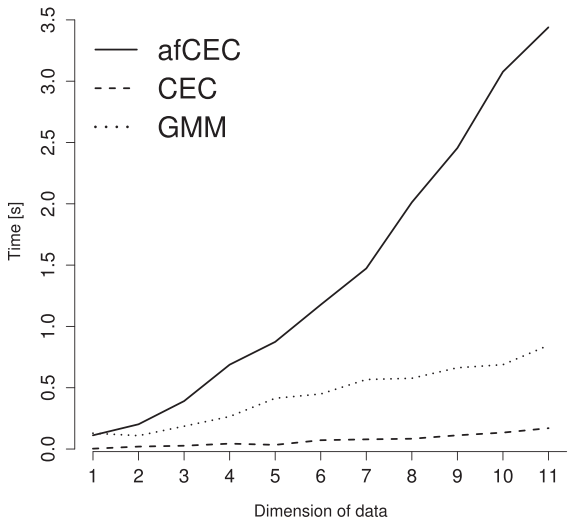
and GMM from R package **Rmixmod** (Lebret et al., 2015). We varied the number of data set instances and the dimension of data, see Fig. 7. In the case of acaGMM we do not have the original author's implementation. Since our implementation of acaGMM gives much worse results then afCEC, we decided to not provide it in the comparison, since it can be caused by the non-optimality of our implementation. For this purpose a simple mouse-like set with 3 well separated Gaussian component was considered.

One can observe that afCECgives similar results to **CEC** and **Rmixmod**. In the case of afCECwe have to apply a regression in each cluster therefore it is slightly worst.

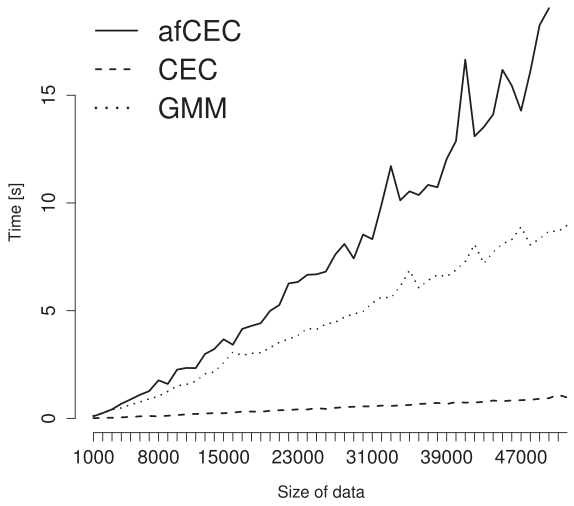
5. Experiments

In this section, we present a comparison of the afCECwith density based methods AcaGMM, GMM, CEC. Since AcaGMM is not a density model, the log-likelihood function is not well-defined. Nevertheless, by inputing the Jacobian of the AcaGMM transformation, we obtain a valid probability distribution. This modification was applied in order to compare the methods.

To compare the results, we use the standard Bayesian Information Criterion (BIC): $BIC = -2LL + k\log(n)$, and Akaike Information Criterion (AIC): $ACI = -2LL + 2k$, where k is the number of parameters in the model, n is the number of points, and LL is a maximized value of the log-likelihood function. Consequently, we need a number of parameters which are used in each model. In the case of $\mathbb{R}^2$, AcaGMM uses two scalars for mean, three scalars for covariance matrix, two scalars for parabola, and one for the local coordinate system (obtained by PCA). On the other hand, in afCECwe do not need scalars for the local coordinate system. Consequently,



(a) Comparison of computational efficiency between afCEC, CEC and GMM in the case of data with different dimensions.



(b) Comparison of computational efficiency between afCEC, CEC and GMM in the case of data with different number of elements.

Fig. 7. Comparison of computational efficiency between afCEC, CEC and GMM.

afCECuses two scalars for mean, three scalars for covariance matrix, and two scalars for parabola.[7]

5.1. Synthetic data set

Let us start from a synthetic data set. First, we report the results of afCEC, AcaGMM, CEC and GMM in the case of a circle-type set, see Fig. 8. Fig. 8(e) shows how the log-likelihood function changes when the number of clusters increases from 1 to 10. A similar relation, with respect to the number of parameters[8] is presented in Fig. 8(f). For a similar values of the log-likelihood function, we need two clusters in afCECand AcaGMM and four in GMM and CEC, see Fig. 8. In such a case, the BIC criterion shows that algorithms which use the curved densities model fit data better using a smaller number of parameters.

A similar situation can be observed in the more complex case of a spiral-type set, see Fig. 9. In Table 1, the mean (with standard deviation) and maximum value of Log-likelihood for 100 initializations of algorithms are shown. As we see for similar values of the log-likelihood function, we have to use nine clusters for afCECand AcaGMM and fourteen for GMM and CEC. We present the comparison of algorithms by using BIC and AIC with similar values of the log-likelihood function in Table 2.

Algorithms which are able to adapt to curve type structures (AcaGMM, afCEC) fit data better. More precisely, the log-likelihood function takes a larger value with the same number of parameters, see Figs. 8(f) and 9(f). Since Log-likelihood increases with the number of classes, we use BIC criterion, which takes into account the number of parameters. In the case of AcaGMM and afCEC, we obtain the optimal value of BIC after about 4–6 iterations. In conclusion, AcaGMM and afCECfit data better in that they yield a higher value of the Log-likelihood function while requiring a lower number of parameters.

Algorithms AcaGMM and afCECgive a comparable value of log-likelihood, see Figs. 8(e) and 9(e). Nevertheless, afCECuses less parameters, see Figs. 8(f) and 9(f). Moreover, the strong theoretical background of this method guarantees that the cost function decreases in each iteration. Consequently, we obtain a simple stop condition for our method.

Chinese characters mainly consist of straight-line strokes (horizontal, vertical) and curve strokes (slash, backslash and many types of hooks). GMM has already been employed for analyzing the structure of Chinese characters and has achieved commendable performance Zhang, Zhang, and Yi (2004). However, some lines extracted by GMM may be too short, and it is quite difficult to join these short lines to form semantic strokes due to the ambiguity of joining them together. This problem becomes more serious when analyzing handwritten characters by GMM, and this was the motivation to use AcaGMM to represent Chinese characters. In Table 3, we present a comparison of afCEC, AcaGMM, GMM, and CEC for Chinese and Latin characters: 犬 (dog), 乞 (beg), 父 (father), 仉 (mother), 火 (fire), 主 (master), b, R, S. The number of clusters has been determined so as to obtain a similar value of log-likelihood function.

At the end of this subsection we present how our method works in the case of segmentation 3D objects. The effect of afCECon three objects (Bronstein, Bronstein, & Kimmel, 2006; 2008) is shown in Fig. 10.

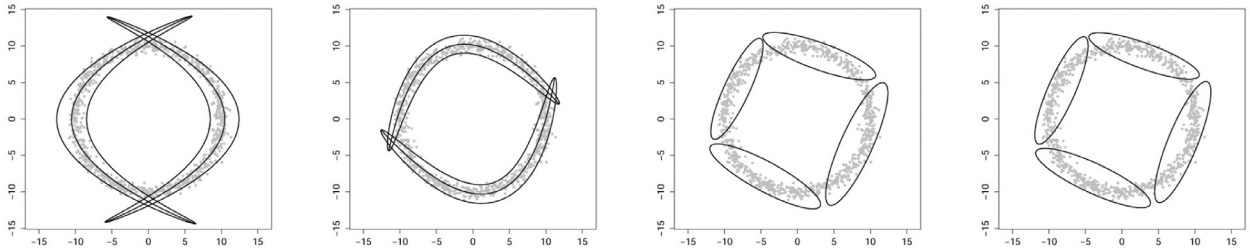
5.2. Data from the UCI repository

In this subsection we compare k-means, EM, CEC and afCEC with respect to Rand and Jaccard indexes, see Table 4. In the case of data sets of dimension higher then three, due to computational profitability, we used smaller class of quadratic polynomials of the type

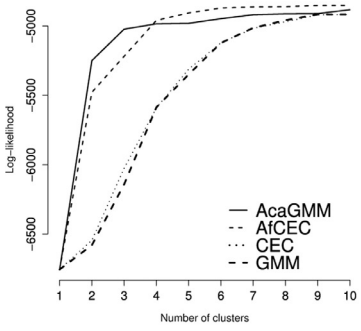
$$f(x_1,\ldots,x_{d-1}) = \sum_{i=1}^{d-1} a_i x_i^2 + \sum_{i=1}^{d-1} b_i x_i + c,$$

[7] It should be emphasized that in afCEC, we need to remember which coordinate is the dependent one. This parameter is discrete, so we do not consider it in our investigation.

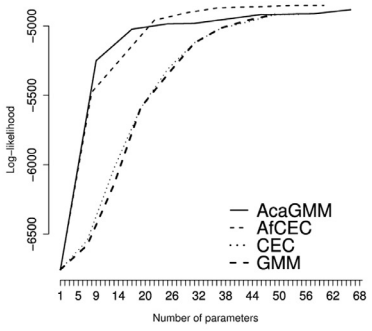
[8] Plots which present a relation between log-likelihood functions and the number of parameters were constructed by linear approximation of known values of the function.



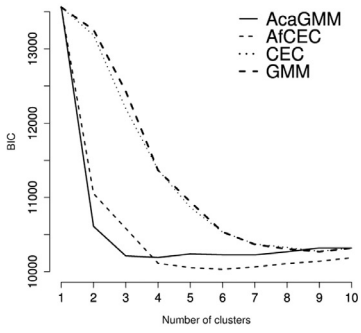
(a) afCEC (b) AcaGMM (c) GMM. (d) CEC



(e) Evolution of Log-likelihood function when the number of clusters increases from 1 to 10.

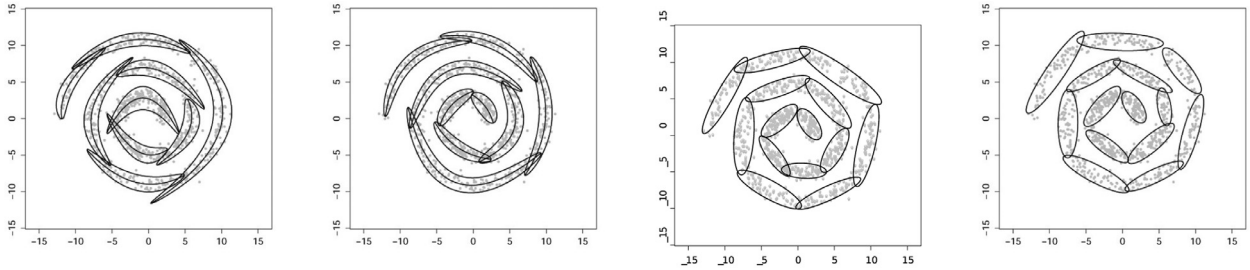


(f) Evolution of Log-likelihood function when the number of parameters increases from 1 to 80.

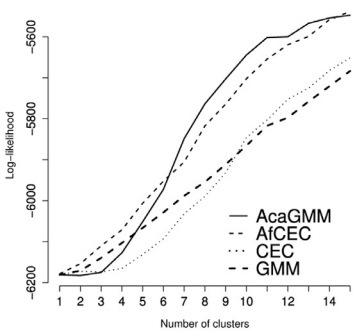


(g) Evolution of BIC function when the number of cluster increases from 1 to 10.

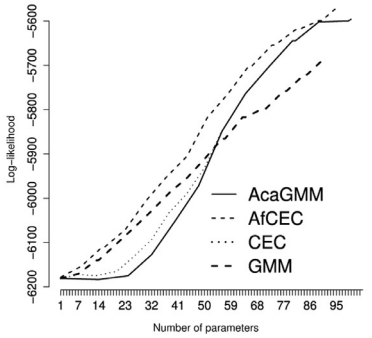
Fig. 8. Results of afCEC, AcaGMM, CEC, and GMM in the case of a circle-type set.



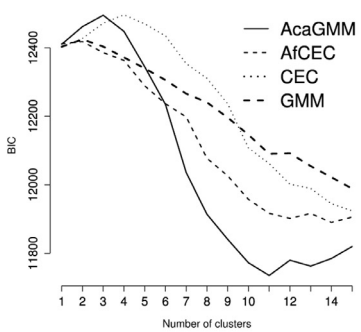
(a) AfCEC (b) AcaGMM (c) GMM (d) CEC



(e) Evolution of Log-likelihood function when the number of clusters increases from 1 to 15.



(f) Evolution of Log-likelihood function when the number of parameters increases from 1 to 120.



(g) Evolution of BIC function when the number of cluster increases from 1 to 10.

Fig. 9. Results of afCEC, AcaGMM, CEC, and GMM in the case of a spiral-type set.

Table 1
Comparison of afCEC, AcaGMM, CEC, and GMM in the case of a spiral-type set, see Fig. 9.

	afCEC			AcaGMM			GMM			CEC		
	NP	mean LL	max LL	NP	mean LL	max LL	NP	mean LL	max LL	NP	mean LL	max LL
1	7	−6178,30 ± 0.000	−6178,30	8	−6180,86 ± 0.000	−6180,86	6	−6180,68 ± 0.00	−6180,68	6	−6180,68 ± 0.000	−6180,68
2	14	−6153,61 ± 18.17	−6069,15	16	−6182,41 ± 6.237	−6104,58	12	−6172,16 ± 0.619	−6157,13	12	−6170,67 ± 13.88	−6127,52
3	21	−6109,99 ± 14.29	−6012,19	24	−6174,88 ± 11.23	−6068,96	18	−6173,61 ± 4.611	−6128,87	18	−6139,47 ± 15.86	−6088,73
4	28	−6070,96 ± 27.36	−5924,87	32	−6127,14 ± 26.67	−5987,66	24	−6165,16 ± 6.619	−6062,66	24	−6102,95 ± 12.27	−6041,99
5	35	−6006,17 ± 40.55	−5868,56	40	−6051,35 ± 31.21	−5836,19	30	−6131,26 ± 11.51	−6026,12	30	−6066,26 ± 13.92	−5989,85
6	42	−5952,44 ± 39.29	−5713,61	48	−5972,21 ± 42.12	−5667,10	36	−6093,05 ± 16.54	−5990,69	36	−6028,42 ± 18.96	−5953,28
7	49	−5905,57 ± 52.18	−5675,39	56	−5848,57 ± 37.63	−5558,34	42	−6031,69 ± 20.44	−5930,99	42	−5987,86 ± 20.74	−5882,36
8	56	−5817,57 ± 65.33	−5612,98	64	−5763,63 ± 37.77	−5511,39	48	−5989,59 ± 26.67	−5868,69	48	−5954,45 ± 29.51	−5865,29
9	63	−5764,29 ± 73.84	−5509,13	72	−5702,82 ± 46.09	−5482,30	54	−5931,64 ± 25.07	−5814,95	54	−5911,52 ± 29.46	−5804,07
10	70	−5702,46 ± 66.64	−5494,73	80	−5644,46 ± 55.69	−5460,11	60	−5846,39 ± 24.83	−5741,61	60	−5865,69 ± 27.32	−5766,89
11	77	−5654,33 ± 85.60	−5441,22	88	−5601,65 ± 58.26	−5435,63	66	−5802,84 ± 27.46	−5689,48	66	−5817,09 ± 38.18	−5713,91
12	84	−5619,46 ± 70.28	−5410,99	96	−5599,81 ± 64.26	−5448,04	72	−5752,16 ± 30.18	−5636,58	72	−5797,22 ± 35.98	−5664,69
13	91	−5598,93 ± 84.31	−5430,61	104	−5566,99 ± 60.39	−5423,52	78	−5725,24 ± 31.85	−5609,44	78	−5757,77 ± 40.37	−5623,56
14	98	−5558,18 ± 89.70	−5384,77	112	−5553,93 ± 69.19	−5420,68	84	−5682,13 ± 31.51	−5542,43	84	−5720,74 ± 41.81	−5563,87
15	105	−5538,44 ± 96.44	−5392,29	120	−5547,13 ± 66.10	−5431,19	90	−5651,20 ± 32.67	−5554,23	90	−5683,41 ± 45.22	−5555,02

Table 2
Comparison of afCEC, AcaGMM, CEC, and GMM in the case of a spiral-type set, see Fig. 9.

Algorithms	Number of clusters	Number of parameters	Log-likelihood	BIC	AIC
afCEC	9	9·7 = 63	−5508.83	11452.85	11143.66
AcaGMM	9	9·8 = 72	−5497.11	11491.58	11138.22
GMM	14	14·6 = 84	−5520.96	11622.17	11209.92
CEC	14	14·6 = 84	−5510.09	11600.44	11188.18

Table 3
Comparison of the afCEC, AcaGMM, CEC and GMM methods for Chinese and Latin characters.

	afCEC			AcaGMM			GMM			CEC		
	NC	LL	BIC	NC	LL	BIC	NC	LL	BIC	NC	LL	BIC
犬	5	1148.57	−2071.93	5	1030.02	−1802.66	7	1060.26	−1850.18	7	1015.29	−1760.33
乞	5	1000.78	−1770.42	5	959.71	−1655.26	7	1170.85	−2064.33	7	1175.96	−2074.55
父	4	1009.02	−1836.69	4	880.32	−1553.38	5	824.51	−1454.71	5	811.76	−1429.21
父	4	1009.02	−1836.69	4	880.32	−1553.38	6	1027.55	−1821.93	6	1032.92	−1832.67
仉	6	1329.01	−2372.74	6	1272.74	−2219.45	8	1364.94	−2403.85	8	1422.27	−2518.51
火	4	1045.53	−1911.53	4	921.65	−1638.12	5	900.25	−1608.15	5	902.12	−1611.89
火	4	1045.53	−1911.53	4	921.65	−1638.12	6	1017.13	−1803.44	6	1018.31	−1805.79
主	5	1011.27	−1794.47	5	962.93	−1665.21	7	1079.99	−1840.69	7	1181.03	−2042.77
b	3	2660.87	−5158.99	3	2738.24	−5290.49	4	2686.59	−5187.19	4	2678.49	−5170.99
R	3	1911.73	−3652.67	3	1578.61	−2962.04	4	1996.56	−3797.94	4	1989.31	−3783.43
S	3	1883.88	−3604.71	3	1907.83	−3629.32	4	1875.93	−3565.52	4	1866.01	−3545.68

instead of the class of all quadratic polynomials

$$f(x_1,\ldots,x_{d-1}) = \sum_{i=1}^{d-1}\sum_{j=1}^{d-1} a_{ij}x_ix_j + \sum_{i=1}^{d-1} b_ix_i + c.$$

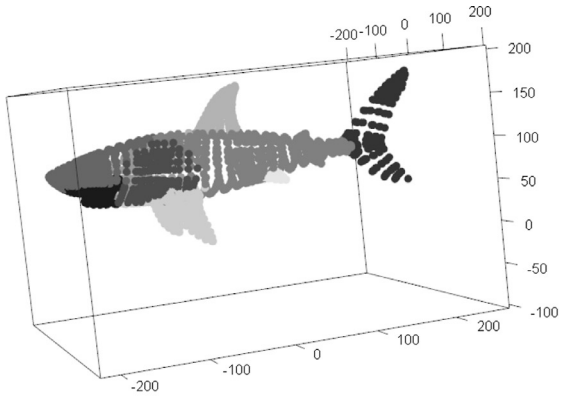
This allows to fit less parameters in each step, which results in smaller risk of overfitting and helps to effectively cluster higher dimensional data.

In most cases afCEC gives better results than classical algorithms, which means that data sets represent curve (or manifold) type structures.

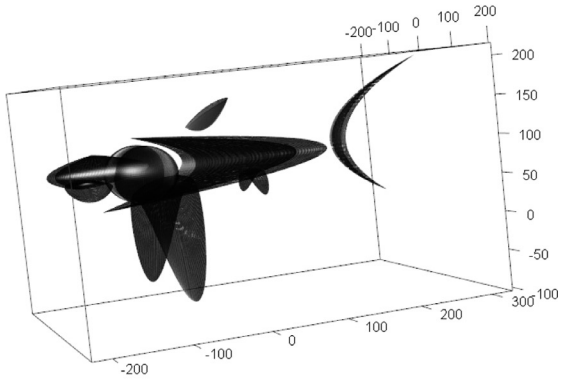
5.3. Data from the monitoring systems of wind turbines

At the end of this section, we examine the method on data from the monitoring systems of wind turbines (Barszcz, Bielecka, Bielecki, & Wójcik, 2011; Barszcz, Bielecki, & Wójcik, 2010; Bielecki, Barszcz, Wójcik, & Bielecka, 2013, 2014). The growing number of that type of systems necessitates an analysis of gigabytes of the data obtained every day. Apart from the development of several advanced diagnostic methods for this type of machinery, there is a need for a group of methods that can act as “early warning tools”. The idea of this approach could be based on a data driven algorithm that would decide on the similarity of current data to data that are already known.

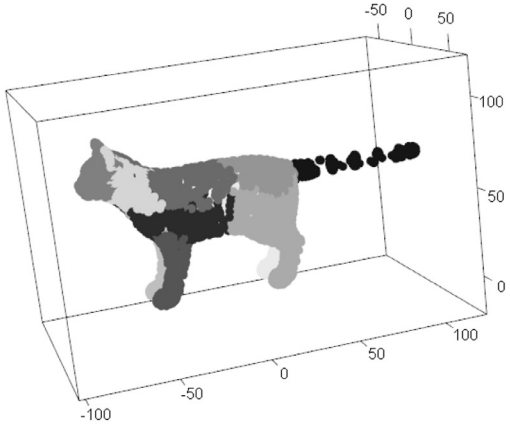
Using density-based clustering algorithm data from a turbine which is in good condition could be described by density estimation. If there are data points which do not match the density, this means that an unknown operational state of the turbine has occurred and an expert should be informed about the situation. Classical CEC and GMM algorithms have been used in such cases (Spurek, Wójcik, & Tabor, 2015), therefore we verified afCEC in a similar situation. Let us consider data in $\mathbb{R}^4$ covering the period from 04/18/2014 to 04/29/2014, recorded every 1 second by an online monitoring system. The data set contained only the basic values that define the operational state of a turbine: wind speed, rotational speed of the rotor, and the AC/DC power generated by the turbine. Negative values of AC power indicated that the power was absorbed. The recorded data were not averaged. The data set included 985,837 measurements. The data where situated in s lower-dimensional subspace and were strongly correlated in one direc-



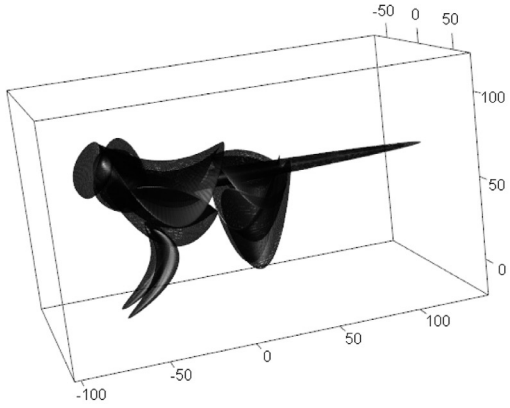
(a) The original data with clusters marked by different colors.



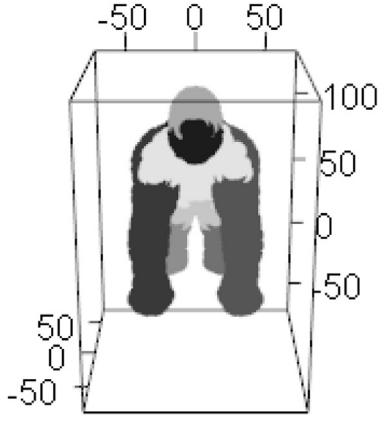
(b) The 3D level sets of curved Gaussian, which describe clusters.



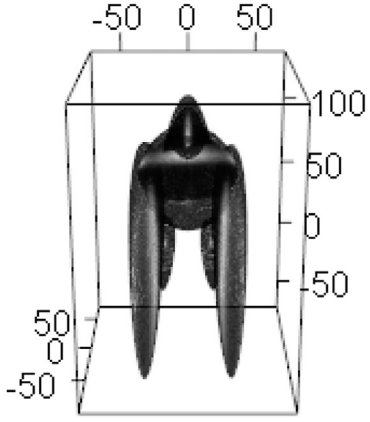
(c) The original data with clusters marked by different colors.



(d) The 3D level sets of curved Gaussian, which describe clusters.



(e) The original data with clusters marked by different colors.



(f) The 3D level sets of curved Gaussian, which describe clusters.

Fig. 10. Result of afCECalgorithm in the case of a 3D data sets.

tion, as the last eigenvalue of the covariance matrix where essentially smaller than the first three: 350102.91, 2006.03, 126.14, 0.76. Therefore, we used PCA (Principal Component Analysis) for extracting the three most important dimensions.

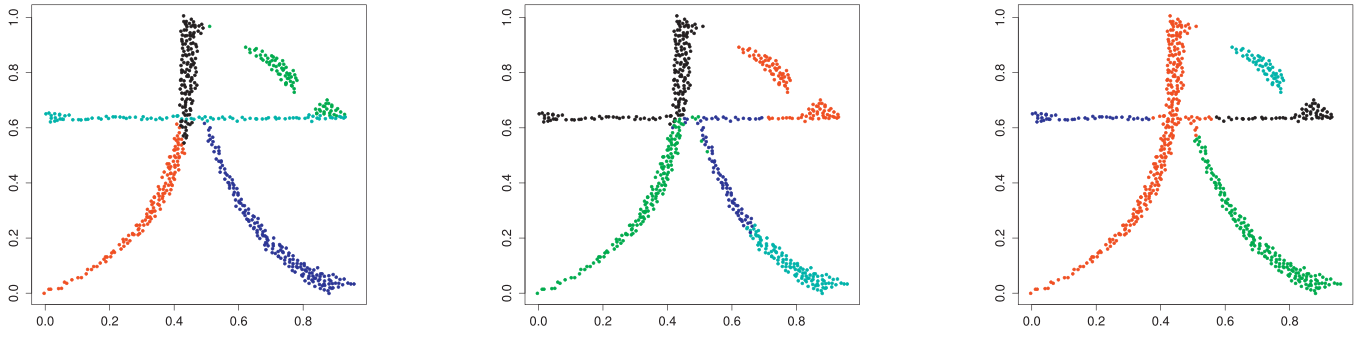
The results of the experiment are included in Table 5. The afCEC method gives a better value of the Log-Likelihood function. More precisely we need 8–10 clusters in GMM and CEC to obtain a similar approximation as in afCEC with 2–4 groups.

Table 4
Comparison between *k*-means, EM, CEC ans afCEC methods by using Rand index and Jaccard index.

	Nr.	Dimension	Rand index			
	of clusters	of data	*k*-means	EM	CEC	afCEC
Wine	3	13	0.7186568	0.878055	0.889164	**0.9039548**
Yeast	10	8	**0.7506702**	0.7295158	0.7306845	0.722322
Diabetes	2	8	0.520715	0.5294069	0.5306389	**0.5450348**
Iris dataset	3	4	0.8797315	**0.9656376**	**0.9656376**	0.9363758
	Nr.	Dimension	Jaccard index			
	of clusters	of data	*k*-means	EM	CEC	afCEC
Wine	3	13	0.4119676	0.6946431	0.7159128	**0.7497933**
Yeast	10	8	0.1541658	0.1821047	0.1808168	**0.2318273**
Diabetes	2	8	0.4575517	0.407901	0.4045691	**0.5450348**
Iris dataset	3	4	0.6958588	**0.9009032**	**0.9009032**	0.8133368

Table 5
Comparison of afCEC, CEC, and GMM on data from the monitoring systems of wind turbines.

Number	afCEC			GMM			CEC		
of cluster	NP	LL	BIC	NP	LL	BIC	NP	LL	BIC
1	16	508,905,829	−1017630	10	491,653,641	−983193	10	491,653,641	−983193
2	32	1,991,410,353	−3982457	20	773,365,127	−1546503	20	734,563,127	−1468899
3	48	2,510,879,469	−5021214	30	834,738,072	−1669135	30	835,448,634	−1670557
4	64	2,615,773,528	−5230820	40	927,726,228	−1854998	40	966,423,228	−1932392
5	80	2,794,697,445	−5588486	50	1,046,249,713	−2091931	50	1,125,648,785	−2250730
6	96	2,885,364,198	−5769638	60	1,049,321,974	−2097962	60	1,168,945,133	−2337209
7	112	2,887,169,707	−5773067	70	1,232,837,612	−2464880	70	1,197,218,483	−2393642
8	128	2,966,253,756	−5931054	80	1,237,747,278	−2474586	80	1,234,625,938	−2468343
9	144	2,982,195,923	−5962756	90	1,291,171,007	−2581320	90	1,270,913,094	−2540804
10	160	3,003,720,585	−6005624	100	1,324,271,861	−2647408	100	1,334,241,112	−2667346



(a) CEC (b) kkmena (c) Spectral clustering

Fig. 11. The effect of clustering Chinese character by CEC, kkmena and spectral clustering.

5.4. Comparison with non-density based methods

Now we present comparison between afCECand classical approaches dedicated for clustering of nonlinear datasets: kkmeans (Li et al., 2008), SOM (Kohonen, 2001), and spectral clustering (Ng et al., 2002) (see Fig. 11).

Kernel methods and self-organizing maps can be used as a preprocessing for classical clustering methods. In such a way spectral clustering methods was constructed (Chi & Yang, 2006; Li et al., 2008; Ng et al., 2002). The classical kernel *k*-means (Li et al., 2008) is equivalent to KPCA prior to the conventional *k*-means algorithm. Most of kernel methods consist of two steps: an embedding into a feature space and a classical clustering method used on the data transformed to feature space. Therefore spectral methods are typically time consuming and use large number of parameters.

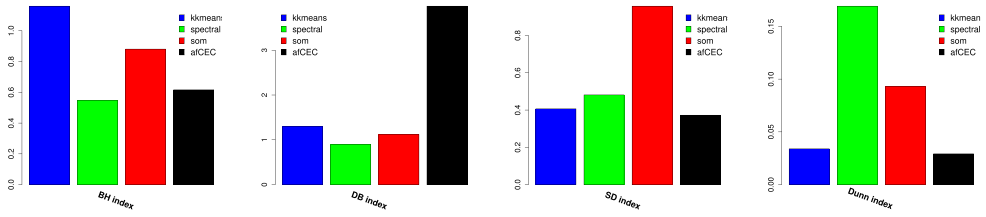
In the case of non-density based method we use many internal quality indexes which have been proposed by various authors in order to determine an optimal clustering. In our work we use BH index introduced by Ball and Hall (1965), DB index proposed by Davies and Bouldin (1979), SD index (Halkidi, Batistakis, & Vazirgiannis, 2001) and Dunn index (Dunn, 1974).

The first two indexes measure internal consistence of clusters, the next two describe how the clusters are separated. As we see in Fig. 12 afCECmethod gives similar results to other approaches.

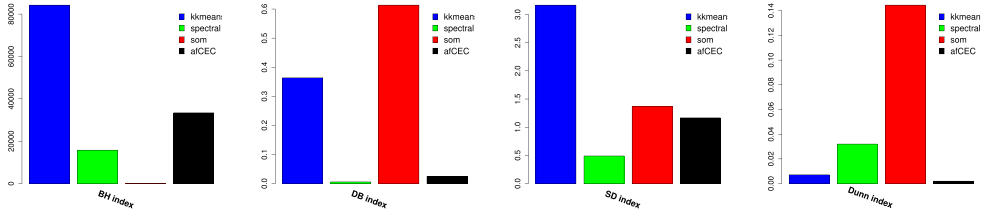
6. Conclusion

In this paper, the afCEC method for clustering curved data, which uses generalized Gaussian distributions in curvilinear coordinate systems, was presented. The afCEC method has a strong theoretical background, and in particular, the cost function decreases in each iteration (Observation 4.1). Moreover, afCEC can be used as a density estimation model. Since afCEC is an implementation of the Cross-Entropy clustering approach the method reduces on-line unnecessary clusters.

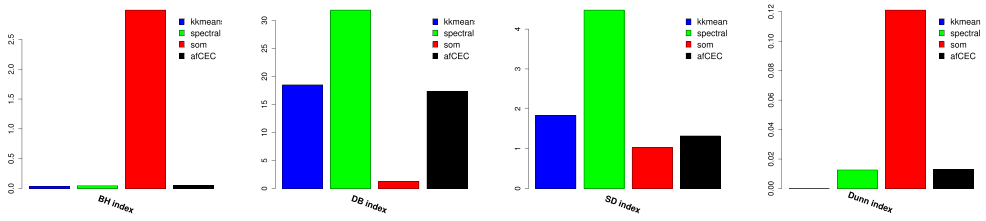
In practice, the approach gives essentially better results than linear models like GMM or CEC, since we obtain a similar level of the Log-likelihood function by using a smaller number of pa-



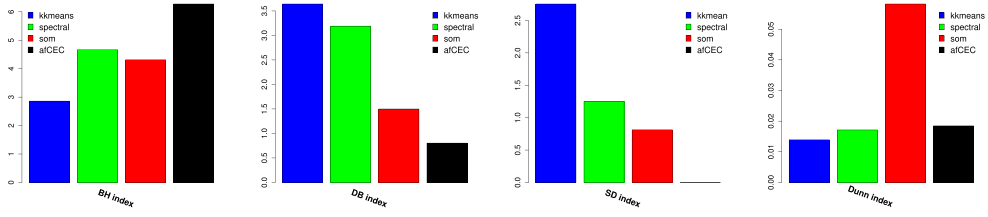
(a) Iris dataset.



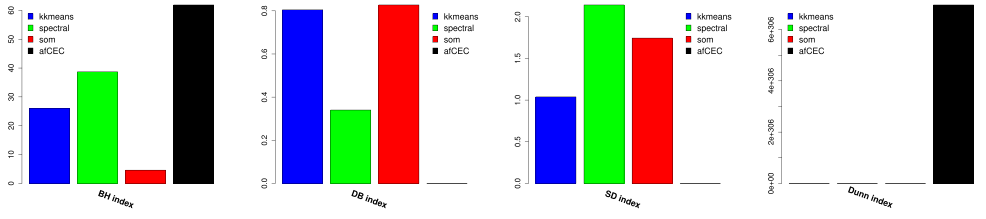
(b) Wine dataset.



(c) Yeast dataset.



(d) Glass dataset.



(e) Breast dataset.

Fig. 12. The effect of clustering Chinese character by CEC, kkmena and spectral clustering.

rameters to describe the model. On the other hand, the results are similar of AcaGMM when we restrict the data to two dimensions and use the quadratic function as the baseline.

Acknowledgment

The work of P. Spurek was cofounded by the European Union from resources of the European Social Fund. Project PO KL "Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00. The work of J. Tabor was supported by the National Centre of Science (Poland) grant no. 2014/13/B/ST6/01792. The work of K. Byrski was supported by the National Centre of Science (Poland) grant no. 2015/19/D/ST6/01472.

Appendix A. AcaGMM Gaussian model

As it was previously mentioned, AcaGMM does not use densities. More precisely, the Jacobian of the transformation was not taken into consideration. However, the EM procedure, which was used in AcaGMM, works with probability distributions. Therefore, from the theoretical point of view the above procedure is incorrect. Moreover, if we want to compare our method by using of the Log-likelihood function we need densities.

Let us start from numerical integration of the original AcaGMM function and of the model rescaled by Jacobian correction. The Simpson method James, Smith, and Wolford (1985), on the square $[-5,5]\times[-5,5]$ with 50000 segments was used. The integral in the case of AcaGMM is equal to 1.038. After correction we obtain 1 (with a precision of 10^4).

Let us consider situation of the AcaGMM model. Suppose X and Y are zero mean independent Gaussian distributions with variances σ_1, σ_2:

$$N_{XY}(x,y)=\frac{1}{\sqrt{2\pi}\sigma_1\sigma_2}\exp\left(-\frac{x^2+y^2}{2\sigma_1\sigma_2}\right).$$

Moreover, let

$$Z=g(X,Y),\quad W=h(X,Y),$$

where $g, h\in\mathcal{C}(\mathbb{R}^2,\mathbb{R})$. Let $J(x, y)$ represent the Jacobian of the original transformation

$$J(x,y)=\det\begin{bmatrix}\frac{\partial g(x,y)}{\partial x} & \frac{\partial g(x,y)}{\partial y}\\ \frac{\partial h(x,y)}{\partial x} & \frac{\partial h(x,y)}{\partial y}\end{bmatrix}.$$

In such a case, we have

$$N_{ZW}(z,w)=\sum_{\{(x,y)\in\mathbb{R}^2:(g(x,y),h(x,y))=(z,w)\}}\frac{N_{XY}(x,y)}{|J(x,y)|}.$$

Let us consider the function f expressed as parametric equation $f:=\{(x(t),y(t)):t\in\mathbb{R}\}$ (in the case of AcaGMM it is a parabola). Using the formula from Zhang et al. (2005, Table 1.) we obtain the orthogonal projection $(x(t_0), y(t_0))$ of point (p_1, p_2) on curve f:

$$t_0=p_f(p_1,p_2)$$

$$=\begin{cases}\sqrt[3]{R+\sqrt{D}}+\sqrt[3]{R-\sqrt{D}} & D>0\\ 0,\ 0,\ 0 & D=0, Q=R=0\\ 2\sqrt{-Q},\ -\sqrt{-Q},\ -\sqrt{-Q} & D=0, Q\neq 0, R\neq 0\\ 2\sqrt{-Q}\ cos(\frac{\phi+2i\pi}{3}), i=0,1,2, & D<0\\ \text{where } \phi=acos\left(\frac{R}{\sqrt{-Q^3}}\right)\end{cases}$$

where $Q=\frac{1-2ap_2}{6a^2}$, $R=\frac{p_1}{4a^2}$ and $D=Q^3+R^2$.

On the other hand, the arc length of f between zero and $(x(t_0), y(t_0))$ (Zhang et al., 2005, Formula (10)) is given by

$$l(t_0)=\frac{1}{2}|t_0|\sqrt{1+4a^2t_0^2}+\frac{1}{4a}\ln\left(2|a|t_0+\sqrt{1+4a^2t_0^2}\right).$$

Consequently, we have

$$g^{-1}(p_1,p_2)=\|(x(t_0),y(t_0))-(p_1,p_2)\|,$$

$$h^{-1}(p_1,p_2)=l(t_0),$$

where $t_0=p_f(p_1,p_2)$.

Our goal is to determine the Jacobian of our transformation, see Fig. 13. Let us consider an arbitrary small neighborhood of $(x(t_0), y(t_0))$. In such a case, the local curvature of f at $(x(t_0), y(t_0))$ is the same as the curvature of the osculating circle[9] at $(x(t_0), y(t_0))$.

The radius of curvature in the case of parametric form of curve is given by

$$r=\frac{(x'^2+y'^2)^{\frac{3}{2}}}{x'y''-y'x''}.$$

Consequently, our goal is to determinate how a set is changing under the influence of the transformation, see Fig. 14.

A small square neighborhood of the point (p_1, p_2) is mapped to a trapezoid (asymptotically when a size of square converges to zero). This operation is showed in Fig. 14. It is easy to see that the square area changes linearly depending on the distance p. If we consider the situation where $p=r$, we obtain that our square is collapsed to a point. Consequently, for points above the curve Jacobian is asymptotically proportional to

$$\frac{r-p}{r}=1-\frac{p}{r}.$$

In a natural way, if a point (p_1, p_2) is under the curve, the square area is increasing under the influence of the transformation. Therefore, the Jacobian is asymptotically equal to

$$\frac{r+p}{r}=1+\frac{p}{r}.$$

Now we have the formula for the Jacobian of AcaGMM transformation, but it depends on the relation between a point and its orthogonal projection. More precisely, we have to verify which formula should be used (or equivalently on which side of parabola a point is found), see Fig. 15.

We can easily verify where the point (p_1, p_2) is in relation to the orthogonal projection $(x(t_0), y(t_0))$ by checking the orientation of a basis containing the normal vector $(p_1, p_2)-(x(t_0), y(t_0))$ and

[9] In differential geometry of curves, the osculating circle of a sufficiently smooth plane curve at a given point p on the curve has been traditionally defined as the circle passing through p and a pair of additional points on the curve infinitesimally close to p. Its center lies on the inner normal line, and its curvature is the same as that of the given curve at that point. This circle, which is the one among all tangent circles at the given point that approaches the curve most tightly, was named circulus osculans (Latin for "kissing circle") by Leibniz.

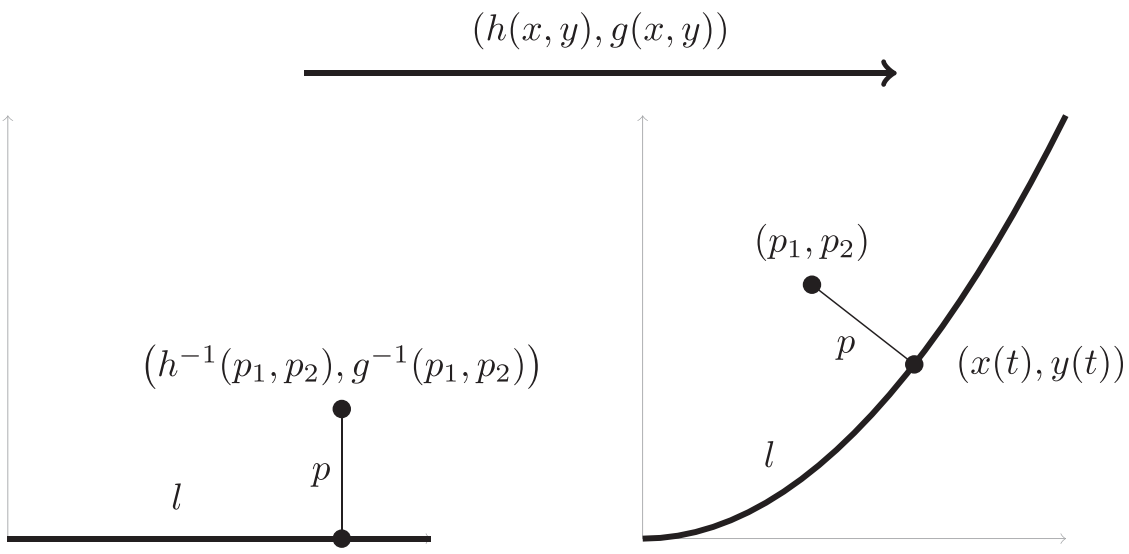


Fig. 13. The transformation used in AcaGMM.

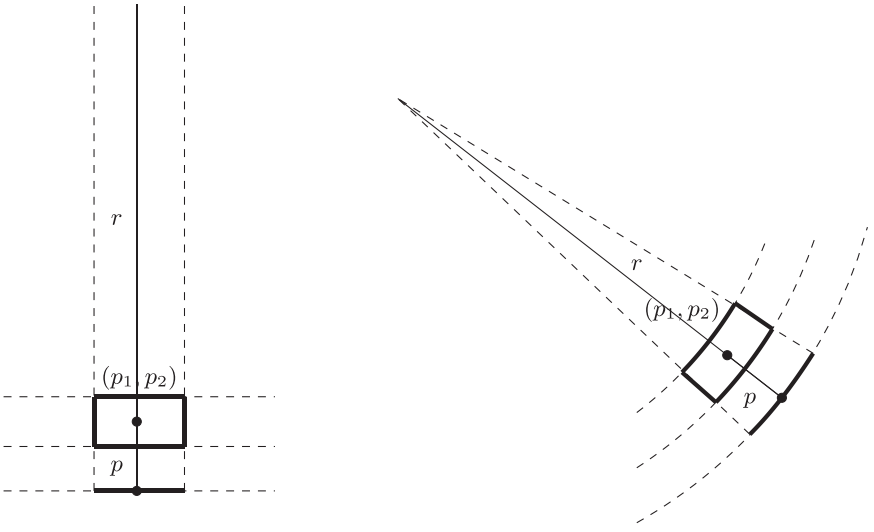


Fig. 14. Transformation of a square neighborhood of a point (p_1, p_2) under the influence of the AcaGMM function.

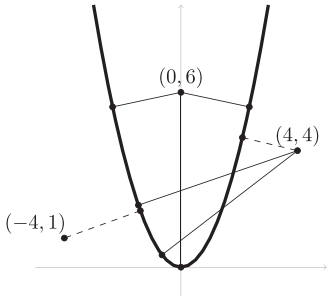


Fig. 15. Position of the point and its orthogonal projection on the parabola $f(x) = x^2$. The distance between point and his orthogonal projection, when it is situated above the curve is marked by a solid line. On the other hand, if the relationship is reversed, we mark the projection by a dashed line.

the tangent vector $(x'(t_0), y'(t_0))$ at a point $(x(t_0), y(t_0))$. Consequently, we have to verify the sign of the determinant

$$\det\left(\begin{bmatrix} p_1 - x(t_0) & x'(t_0) \\ p_2 - y(t_0) & x'(t_0) \end{bmatrix}\right).$$

Appendix B. Proof of Theorem 3.1

Let us start with simple Lemma.

Lemma B.1. *Let* $\mathrm{m} \in \mathbb{R}^d$, $\Sigma_{\hat{l}} \in \mathcal{M}_{d-1}(\mathbb{R})$, $\Sigma_l > 0$ *and* $\mathrm{v} \in \mathbb{R}^{d-1}$ *be given. Then for* $A = \begin{bmatrix} I_{d-1} & 0 \\ \mathrm{v}^T & -1 \end{bmatrix}$ *we have*

$$N\left(A\mathrm{m}, A\begin{bmatrix} \Sigma_{\hat{l}} & 0 \\ 0 & \Sigma_l \end{bmatrix}A^T\right)(\mathrm{x}) = N(\mathrm{m}, \Sigma_{\hat{l}}, \Sigma_l, f)(\mathrm{x}),$$

where $f : \mathbb{R}^{d-1} \to \mathbb{R}$ *such, that* $f(\mathrm{x}) = \mathrm{v}^T \cdot \mathrm{x}$.

Proof. Assume, without a loss of generality, that $l = d$. Let us denote $\Sigma = \begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ 0 & \Sigma_d \end{bmatrix}$ and $\mathrm{m} = [\mathrm{m}_{\hat{d}}, m_d]^T$, then we have

$$N(A\mathrm{m}, A\Sigma A^T)(\mathrm{x}) = N\left(\begin{bmatrix} I_{d-1} & 0 \\ \mathrm{v}^T & -1 \end{bmatrix}\begin{bmatrix} \mathrm{m}_{\hat{d}} \\ m_d \end{bmatrix}, \begin{bmatrix} I_{d-1} & 0 \\ \mathrm{v}^T & -1 \end{bmatrix}\begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ 0 & \Sigma_d \end{bmatrix}\begin{bmatrix} I_{d-1} & 0 \\ \mathrm{v}^T & -1 \end{bmatrix}^T\right)(\mathrm{x})$$

$$=N\left(\begin{bmatrix} \mathrm{m}_{\hat{d}} \\ \mathrm{v}^T\mathrm{m}_{\hat{d}} - m_d \end{bmatrix}, \begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ \mathrm{v}^T\Sigma_{\hat{d}} & -\Sigma_d \end{bmatrix}\begin{bmatrix} I_{d-1} & \mathrm{v} \\ 0 & -1 \end{bmatrix}\right)(\mathrm{x})$$

$$=N\left(\begin{bmatrix} \mathrm{m}_{\hat{d}} \\ \mathrm{v}^T\mathrm{m}_{\hat{d}} - m_d \end{bmatrix}, \begin{bmatrix} \Sigma_{\hat{d}} & \Sigma_{\hat{d}}\mathrm{v} \\ \mathrm{v}^T\Sigma_{\hat{d}} & \mathrm{v}^T\Sigma_{\hat{d}}\mathrm{v} + \Sigma_d \end{bmatrix}\right)(\mathrm{x}).$$

It is easy to show that

$$(A\Sigma A^T)^{-1} = \begin{bmatrix} \Sigma_{\hat{d}} & \Sigma_{\hat{d}}\mathrm{v} \\ \mathrm{v}^T\Sigma_{\hat{d}} & \mathrm{v}^T\Sigma_{\hat{d}}\mathrm{v} + \Sigma_d \end{bmatrix}^{-1} =$$

$$= \begin{bmatrix} \Sigma_{\hat{d}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \Sigma_d^{-1}\begin{bmatrix} \mathrm{v}\mathrm{v}^T & -\mathrm{v} \\ -\mathrm{v}^T & 1 \end{bmatrix} = \begin{bmatrix} \Sigma_{\hat{d}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \Sigma_d^{-1}\begin{bmatrix} -\mathrm{v} \\ 1 \end{bmatrix}[-\mathrm{v}^T, 1].$$

Therefore, we have

$$[\mathrm{x}_{\hat{d}}^T, x_d](A\Sigma A^T)^{-1}\begin{bmatrix} \mathrm{x}_{\hat{d}} \\ x_d \end{bmatrix}$$

$$= [\mathrm{x}_{\hat{d}}^T, x_d]\left(\begin{bmatrix} \Sigma_{\hat{d}}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \Sigma_d^{-1}\begin{bmatrix} -\mathrm{v} \\ 1 \end{bmatrix}[-\mathrm{v}^T, 1]\right)^{-1}\begin{bmatrix} \mathrm{x}_{\hat{d}} \\ x_d \end{bmatrix}$$

$$= \mathrm{x}_{\hat{d}}^T\Sigma_{\hat{d}}^{-1}\mathrm{x}_{\hat{d}} + (x_d - \mathrm{x}_{\hat{d}}^T\mathrm{v})\Sigma_d^{-1}(x_d - \mathrm{x}_{\hat{d}}\mathrm{v}^T)$$

$$= [\mathrm{x}_{\hat{d}}^T, x_d - \mathrm{v}^T\mathrm{x}_{\hat{d}}]\begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ 0 & \Sigma_d \end{bmatrix}^{-1}\begin{bmatrix} \mathrm{x}_{\hat{d}} \\ x_d - \mathrm{v}^T\mathrm{x}_{\hat{d}} \end{bmatrix}.$$

As a simple consequence, we obtain the assertion of the Lemma. □

Now we can prove Theorem 3.1.

Proof. Assume without loss of generality that $l = d$. To prove the assertion of Theorem 3.1, we first show the following inclusion:

$\mathcal{G}[\mathcal{F}] \subset \mathcal{G}(\mathbb{R}^d)$.

Let $\mathrm{m} \in \mathbb{R}^d$, $\Sigma = \begin{bmatrix} \Sigma_{\hat{d}} & 0 \\ 0 & \Sigma_d \end{bmatrix}$ (where $\Sigma_{\hat{d}} \in \mathcal{M}_{d-1}$, $\Sigma_d \in \mathbb{R}$), $\mathrm{v} \in \mathbb{R}^{d-1}$ and $f(\mathrm{x}) = \mathrm{v}^T \cdot \mathrm{x}$ be given and let $N(\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f) \in \mathcal{G}[\mathcal{F}]$. Thanks to Lemma Appendix B.1 for $A = \begin{bmatrix} I & 0 \\ \mathrm{v}^T & -1 \end{bmatrix}$, we have

$N(\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f) = N(A\mathrm{m}, A\Sigma A^T) \in \mathcal{G}(\mathbb{R})$.

We now show the opposite inclusion

$\mathcal{G}(\mathbb{R}^d) \subset \mathcal{G}[\mathcal{F}]$.

Let $\Sigma = \begin{bmatrix} \Sigma_{11} & \mathrm{v} \\ \mathrm{v}^T & \Sigma_{22} \end{bmatrix} \in \mathcal{M}_d(\mathbb{R})$ and $\mathrm{m} \in \mathbb{R}^d$ be given and let $N(\mathrm{m}, \Sigma) \in \mathcal{G}(\mathbb{R}^d)$. We put $\Sigma_{\hat{d}} = \Sigma_{11}$, $\Sigma_d = -\mathrm{v}^T\Sigma_{11}^{-1}\mathrm{v} + \Sigma_{22}$, $f(\mathrm{x}) = \Sigma_{11}^{-1}\mathrm{v}^T\mathrm{x}$ and $A = \begin{bmatrix} I & 0 \\ \mathrm{v}^T\Sigma_{11}^{-1} & -1 \end{bmatrix}$. Thanks to Lemma Appendix B.1, we have

$$N\left[A^{-1}\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f\right]$$

$$= N\left(\mathrm{m}, \begin{bmatrix} I & 0 \\ \mathrm{v}^T\Sigma_{11}^{-1} & -1 \end{bmatrix}\begin{bmatrix} \Sigma_{11} & 0 \\ 0 & -\mathrm{v}^T\Sigma_{11}^{-1}\mathrm{v} + \Sigma_{22} \end{bmatrix}\begin{bmatrix} I & \Sigma_{11}^{-1}\mathrm{v} \\ 0 & -1 \end{bmatrix}\right)$$

$$= N\left(\mathrm{m}, \begin{bmatrix} \Sigma_{11} & 0 \\ \mathrm{v}^T & \mathrm{v}^T\Sigma_{11}^{-1}\mathrm{v} - \Sigma_{22} \end{bmatrix}\begin{bmatrix} I & \Sigma_{11}^{-1}\mathrm{v} \\ 0 & -1 \end{bmatrix}\right)$$

$$= N\left(\mathrm{m}, \begin{bmatrix} \Sigma_{11} & \mathrm{v} \\ \mathrm{v}^T & \Sigma_{22} \end{bmatrix}\right).$$

Consequently,

$N(\mathrm{m}, \Sigma) = N\left(A^{-1}\mathrm{m}, \Sigma_{\hat{d}}, \Sigma_d, f\right) \in \mathcal{G}[\mathcal{F}](\mathbb{R})$,

what finished the proof. □

References

Arthur, D., & Vassilvitskii, S. (2007). K-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms, Society for Industrial and Applied Mathematics* (pp. 1027–1035).

Ball, G. H., & Hall, D. J. (1965). ISODATA, a novel method of data analysis and pattern classification. *Technical report*. DTIC Document.

Banfield, J. D., & Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics, 49*, 803–821.

Barszcz, T., Bielecka, M., Bielecki, A., & Wójcik, M. (2011). Wind turbines states classification by a fuzzy-ART neural network with a stereographic projection as a signal normalization. In *Adaptive and natural computing algorithms: Vol. 6594* (pp. 225–234).

Barszcz, T., Bielecki, A., & Wójcik, M. (2010). ART-Type artificial neural networks applications for classification of operational states in wind turbines. In *Artifical intelligence and soft computing: Vol. 6114* (pp. 11–18).

Basu, S., Naphade, M., & Smith, J. R. (2002). A statistical modeling approach to content based retrieval. In *Proceedings of 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP): vol. 4* (pp. IV–4080). IEEE.

Bielecki, A., Barszcz, T., Wójcik, M., & Bielecka, M. (2013). ART-2 Artificial neural networks applications for classification of vibration signals and operational states of wind turbines for intelligent monitoring. *Diagnostyka, 14*(4), 21–26.

Bielecki, A., Barszcz, T., Wójcik, M., & Bielecka, M. (2014). Hybrid system of ART and RBF neural networks for classification of vibration signals and operational states of wind turbines. In *Artificial intelligence and soft computing: Vol. 8467* (pp. 3–11).

Björck, A. (1996). *Numerical methods for least squares problems*. SIAM.

Bronstein, A. M., Bronstein, M. M., & Kimmel, R. (2006). Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing, 28*(5), 1812–1836.

Bronstein, A. M., Bronstein, M. M., & Kimmel, R. (2008). *Numerical geometry of non–rigid shapes*. Springer.

Campbell, J., Fraley, C., Murtagh, F., & Raftery, A. E. (1997). Linear flaw detection in woven textiles using model-based clustering. *Pattern Recognition Letters, 18*(14), 1539–1548.

Cayton, L. (2005). Algorithms for manifold learning. *Technical Report*.

Celeux, G., & Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern recognition, 28*(5), 781–793.

Chi, S.-C., & Yang, C. C. (2006). Integration of ant colony SOM and k-means for clustering analysis. In *Proceedings of International conference on knowledge-based and intelligent information and engineering systems* (pp. 1–8). Springer.

Cover, T., Thomas, J., Wiley, J., et al. (1991). *Elements of information theory*: Vol. 6. Wiley Online Library.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.

Dasgupta, A., & Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association, 93*(441), 294–302.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 2*, 224–227.

Davis-Stober, C., Broomell, S., & Lorenz, F. (2007). Exploratory data analysis with MATLAB. *Psychometrika, 72*(1), 107–108.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics, 4*(1), 95–104.

Figueiredo, M. A. T., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(3), 381–396.

Fraley, C., & Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal, 41*(8), 578–588.

Grünwald, P. D. (2007). *The minimum description length principle*. MIT press.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. *Journal of intelligent information systems, 17*(2–3), 107–145.

Hartigan, J. (1975). *Clustering algorithms*. John Willey and Sons.

Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association, 84*(406), 502–516.

Hayman, E., & Eklundh, J.-O. (2003). Statistical background subtraction for a mobile observer. In *Proceedings of ninth IEEE international conference on computere vision* (pp. 67–74). IEEE.

Hinton, G. E., Dayan, P., & Revow, M. (1997). Modeling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks, 8*(1), 65–74.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery, 2*(3), 283–304.

Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters, 31*(8), 651–666.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Jain, A., Murty, M., & Flynn, P. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

James, M. L., Smith, G. M., & Wolford, J. (1985). *Applied numerical methods for digital computation*: 2. Harper & Row New York.

Jolliffe, I. (2002). Principal component analysis. *Encyclopedia of Statistics in Behavioral Science*.

Ju, Z., & Liu, H. (2011). A unified fuzzy framework for human-hand motion recognition. *IEEE Transactions on Fuzzy Systems, 19*(5), 901–913.

Ju, Z., & Liu, H. (2012). Fuzzy gaussian mixture models. *Pattern Recognition, 45*(3), 1146–1158.

Kamieniecki, K., & Spurek, P. (2014). *CEC: Cross-entropy clustering*. URL http://CRAN.R-project.org/package=CEC, R package version 0.9.2.
Kegl, B. A. (1999). Principal curves: learning, design, and applications, Ph.D. thesis. Citeseer.
Kohonen, T. (1989). *Self-organizing feature maps*. Springer.
Kohonen, T. (2001). *Self-organizing maps*: 30. Springer Science & Business Media.
Krantz, S. G., & Parks, H. R. (2002). *The implicit function theorem: History, theory, and applications*. Springer.
Krawczyk, B., Woźniak, M., & Cyganek, B. (2014). Clustering-based ensembles for one-class classification. *Information Sciences, 264*, 182–195.
Kumar, S., & Hebert, M. (2003). Man-made structure detection in natural images using a causal multiscale random field. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition: Vol. 1* (pp. I–119). IEEE.
Law, M. H., Figueiredo, M. A., & Jain, A. K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(9), 1154–1166.
LeBlanc, M., & Tibshirani, R. (1994). Adaptive principal surfaces. *Journal of the American Statistical Association, 89*(425), 53–64.
Lebret, R., Iovleff, S., Langrognet, F., Biernacki, C., Celeux, G., & Govaert, G. (2015). Rmixmod: The r package of the model-based unsupervised, supervised and semi-supervised classification mixmod library. *Journal of Statistical Software*, In–press.
Levin, M. S. (2015). Combinatorial clustering: Literature review, methods, examples. *Journal of Communications Technology and Electronics, 60*(12), 1403–1428.
Li, J., Li, X., & Tao, D. (2008). KPCA For semantic object extraction in images. *Pattern Recognition, 41*(10), 3244–3250.
Mahajan, M., Nimbhorkar, P., & Varadarajan, K. (2009). The planar k-means problem is NP-hard. In *Proceedings of international workshop on algorithms and computation* (pp. 274–285). Springer.
McKenna, S. J., Raja, Y., & Gong, S. (1999). Tracking colour objects using adaptive mixture models. *Image and vision computing, 17*(3), 225–231.
McLachlan, G., & Krishnan, T. (2007). *The EM algorithm and extensions*: Vol. 382. John Wiley & Sons.
McLachlan, G., & Peel, D. (2004). *Finite mixture models*. John Wiley & Sons.
Moghaddam, B., & Pentland, A. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 696–710.
Mukherjee, S., Feigelson, E. D., Babu, G. J., Murtagh, F., Fraley, C., & Raftery, A. (1998). Three types of gamma-ray bursts. *The Astrophysical Journal, 508*(1), 314.
Narayanan, H., & Mitter, S. (2010). Sample complexity of testing the manifold hypothesis. In *Advances in neural information processing systems* (pp. 1786–1794).
Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems, 2*, 849–856.
Povinelli, R. J., Johnson, M. T., Lindgren, A. C., & Ye, J. (2004). Time series classification using gaussian mixture models of reconstructed phase spaces. *IEEE Transactions on Knowledge and Data Engineering, 16*(6), 779–783.
Samuelsson, J. (2004). Waveform quantization of speech using gaussian mixture models. In *Proceedings of IEEE international conference on acoustics, speech, and signal processing, 2004: vol. 1* (pp. I–165). IEEE.
Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation, 10*(5), 1299–1319.
Śmieja, M. (2015). Weighted approach to general entropy function. *IMA Journal of Mathematical Control and Information, 32*(2), 329–341.
Śmieja, M., & Wiercioch, M. (2016). Constrained clustering with a complex cluster structure. *Advances in Data Analysis and Classification*, 1–26.
Spurek, P. (2017). General split gaussian cross–entropy clustering. *Expert Systems with Applications, 68*, 58–68.
Spurek, P., & Pałka, W. (2016). Clustering of gaussian distributions. In *Proceedings of 2016 international joint conference on neural networks (IJCNN)* (pp. 3346–3353). IEEE.
Spurek, P., & Tabor, J. (2013). The memory center. *Information Sciences, 252*, 132–143.
Spurek, P., Wójcik, M., & Tabor, J. (2015). Cross-entropy clustering approach to one–class classification. In *Proceedings of international conference on artificial intelligence and soft computing* (pp. 481–490). Springer.
Spurek, P., Kamieniecki, K., Tabor, J., Misztal, K., & Śmieja, M. (2016). *R Package CEC*. Elsevier. doi:10.1016/j.neucom.2016.08.118.
Stauffer, C., & Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition, 1999: vol. 2*. IEEE.
Tabor, J., & Spurek, P. (2014). Cross-entropy clustering. *Pattern Recognition, 47*(9), 3046–3059.
Tomczyk, A., Spurek, P., Podgórski, M., Misztal, K., & Tabor, J. (2016). Detection of elongated structures with hierarchical active partitions and CEC-based image representation. In *Proceedings of the 9th international conference on computer recognition systems CORES 2015* (pp. 159–168). Springer.
Valente, F., & Wellekens, C. (2004). Variational bayesian feature selection for gaussian mixture models. In *Proceedings of acoustics, speech, and signal processing, 2004. proceedings.(ICASSP'04). IEEE international conference on: vol. 1* (pp. I–513). IEEE.
Xiong, Z., Chen, Y., Wang, R., & Huang, T. S. (2002). Improved information maximization based face and facial feature detection from real-time video and application in a multi-modal person identification system. In *Proceedings of the 4th IEEE international conference on multimodal interfaces* (p. 511). IEEE Computer Society.
Xu, R., & Wunsch, D. (2009). *Clustering*. Wiley-IEEE Press.
Zhang, B., Zhang, C., & Yi, X. (2004). Competitive EM algorithm for finite mixture models. *Pattern recognition, 37*(1), 131–144.
Zhang, B., Zhang, C., & Yi, X. (2005). Active curve axis gaussian mixture models. *Pattern recognition, 38*(12), 2351–2362.