Schedae Informaticae Vol. 25 (2016): 117–126
doi: 10.4467/20838476SI.16.009.6190

tfml 2017
theoretical foundations
of machine learning, Kraków

Uniform Cross-entropy Clustering

Maciej Brzeski, Przemysław Spurek[1]
Faculty of Mathematics and Computer Science
Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland
e-mail: *maciej.brzeski@doctoral.uj.edu.pl*, *przemyslaw.spurek@uj.edu.pl*

Abstract. Robust mixture models approaches, which use non-normal distributions have recently been upgraded to accommodate data with fixed bounds. In this article we propose a new method based on uniform distributions and Cross-Entropy Clustering (CEC). We combine a simple density model with a clustering method which allows to treat groups separately and estimate parameters in each cluster individually. Consequently, we introduce an effective clustering algorithm which deals with non-normal data.

Keywords: Clustering, Cross-entropy, Uniform distribution

1. Introduction

Clustering plays a basic role in many parts of data engineering, pattern recognition, and image analysis. One of the most important clustering methods is the density approach [1, 2]. Most of such algorithms are based on Gaussian Mixture Model [3], which uses Expectation-maximization (EM) procedure [4]. The mixture components describe individual clusters in the data space. Gaussian components are traditionally successful in detecting elliptic clusters [3–5]. However, groups of a different shapes require a solution with involved components of other distributions.

The growing need for more flexible tools to analyze datasets that exhibit non-normal features, including asymmetry, multimodality, heavy tails, and fixed bounds,

Received: 11 December 2016 / Accepted: 30 December 2016
[1] The work of this author was supported by the National Science Centre (Poland) Grant No. 2015/19/D/ST6/01472.

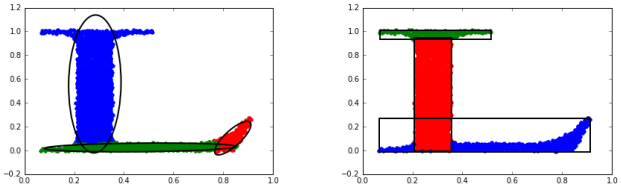
has led to intense development of non-normal model-based methods. The mixture model-based clustering literature has focused on the development of mixture distributions with more flexible parametric components like split distributions [6, 7], skew distributions [8–10] and some other non-elliptical approaches [11–13].

The same situation occurs in the case of clustering non-negative or in some way limited data. To take into account such a feature, components should have a limited support. Therefore, we use the uniform distribution, which well covers clusters in the shape of rectangle. Estimation of data models with the bounded support including uniform ones was studied in various domains: clustering [14], individual state-space and regression models [15, 16] as well as mixture models [17]. In mixture-based clustering approach the challenging task is updating parameters of uniform components. Intuitively, the prior chosen bounds of the uniform distribution are only expandable, but they are not floating (limited support). Therefore, estimating a uniform mixture is very hard.

In this paper we construct a new clustering model Uniform Cross-Entropy Clustering (UCEC), which try to solve these problems. First of all, we use simple multidimensional uniform distribution, see Fig. 1. More precisely, we use uniform pdf for independent variables, which is a product of univariate marginal pdfs, and the distribution will have generally the rectangle support. Furthermore, simpler optimization procedure known as Cross Entropy Clustering (CEC) [18] is used instead of EM.

A goal of CEC is to optimally approximate the scatter of data set $X \subset \mathbb{R}^d$ by the function which is a small modification of EM (for more information see Section 2). It occurs that at the small cost of having a minimally worse density approximation [18], we gain the ease of using more complicated density models. The method is capable of the automatic reduction of unnecessary clusters (contrary to EM each group has its cost). Moreover, we can treat clusters separately which is more effective from a numerical point of view.

This paper is arranged as follows. First the theoretical background of UCEC method is presented. We introduce the cost function which we need to minimize. Moreover, we present three strategies to escape from local minima to reach a better minimum. In the last part numerical experiments are presented.



(a) GMM (b) UCEC

Figure 1. The result of our approach and classical GMM in the case of the L-type dataset.

2. Theoretical background of UCEC

In this section the UCEC method will be presented. First, we introduce the cost function which will be optimized by the algorithm.

Our approach is based on the CEC [18]. Therefore, we start with a short introduction to the method. 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 $f = p_1 f_1 + \ldots + p_k f_k$ optimally approximates the scatter of our data X with respect to the MLE cost function

$$\mathrm{MLE}(f, X) = -\sum_{\mathrm{x} \in X} \ln(p_1 f_1(\mathrm{x}) + \ldots + p_k f_k(\mathrm{x})). \tag{1}$$

A goal of CEC is to minimize the cost function, which is a minor modification of that given in (1) by substituting the sum with the maximum:

$$\mathrm{CEC}(f, X) = -\sum_{\mathrm{x} \in X} \ln(\max(p_1 f_1(\mathrm{x}), \ldots, p_k f_k(\mathrm{x}))). \tag{2}$$

Instead of focusing on the density estimation as its main task, CEC aims directly at the clustering problem. It occurs that a small cost of having a minimally worse density approximation [18], we obtain numerical efficient method. We can often use the Hartigan approach to clustering, which is faster and typically finds better minimums. This is an advantage, roughly speaking, 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. To do so, let it be recalled that by the cross-entropy of data set $X \subset \mathbb{R}^d$ with respect to density f is given by

$$H^{\times}(X \| f) = -\frac{1}{|X|} \sum_{\mathrm{x} \in X} \ln(f(\mathrm{x})).$$

In the case of splitting $X \subset \mathbb{R}^d$ into $X_1, \ldots, X_k$ so that we describe elements of X_i using a function from the family of all multidimensional uniform densities $\mathcal{U}(\mathbb{R}^d)$.

As it was mentioned, we use simple multidimensional uniform distributions. Let us start from one dimensional density

$$U(x; a, b) = \begin{cases} \frac{1}{b-a} & x \in [a, b] \\ 0 & x \notin [a, b] \end{cases},$$

for $a, b \in \mathbb{R}$.

For a dataset $X \subset \mathbb{R}$ the maximum likelihood (ML) estimation of parameters $a, b \in \mathbb{R}$ of uniform distribution is given by maximal and minimal elements of X [19].

In our work we use multidimensional uniform distribution, which is product of univariate marginal pdfs.

Definition 1. *For a vector* $\mathrm{x} \in \mathbb{R}^d$ *the multidimensional uniform distribution is given by*

$$U_d(\mathrm{x}; \mathrm{a}, \mathrm{b}) = \prod_{j=1}^{d} U(x_j; a_j, b_j),$$

for $\mathrm{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$, $\mathrm{a} = [a_1, \ldots, a_d] \in \mathbb{R}^d$, $\mathrm{b} = [b_1, \ldots, b_d] \in \mathbb{R}^d$, *where* $a_i < b_i$ *for* $i = 1, \ldots, d$.

Similar to the one dimensional case the maximum likelihood estimators are given by by maximal and minimal elements of X [19].

Theorem 1. *Let* $X = \{\mathrm{x}_1, \ldots, \mathrm{x}_n\}$ *be a random sample from* $U_d(\mathrm{x}; \mathrm{a}, \mathrm{b})$. *Then the maximum likelihood estimators of* a *and* b *are*

$$\hat{\mathrm{a}} = \min(X) = [\min(X^1), \ldots, \min(X^d)],$$
$$\hat{\mathrm{b}} = \max(X) = [\max(X^1), \ldots, \max(X^d)].$$

The support of uniform density distributions $U_d(\mathrm{x}; \mathrm{a}, \mathrm{b})$ is hyperrectangle

$$supp(U_d(\mathrm{x}; \mathrm{a}, \mathrm{b})) = \{U_d(\mathrm{x}; \mathrm{a}, \mathrm{b}) \neq 0, x \in \mathbb{R}^d\} = [a_1, b_1] \times \ldots \times [a_d, b_d].$$

Therefore, for given uniform distribution $U_d(\mathrm{x}; \mathrm{a}, \mathrm{b})$ volume of his support is equal to $V_{U_d(\mathrm{x};\mathrm{a},\mathrm{b})} = |b_1 - a_1| \cdot \ldots \cdot |b_d - a_d|$. Now we are ready to present the cost function, which will be used in our algorithm

$$E(X_1, \ldots, X_k; \mathcal{U}(\mathbb{R}^d)) = \sum_{i=1}^{k} p_i \cdot \left(-\ln(p_i) + H^{\times}(X_i \| \mathcal{U}(\mathbb{R}^d))\right), \tag{3}$$

where $p_i = \frac{|X_i|}{|X|}$ and $H^{\times}(X_i \| \mathcal{U}(\mathbb{R}^d)) = \inf_{f \in \mathcal{U}(\mathbb{R}^d)} H^{\times}(X_i \| f)$.

The aim of CEC is to split dataset X into subsets X_i which minimize the function given in (3). It is easy to see that in the case of one cluster X, the cross-entropy is equivalent to the log-likelihood function:

$$H^{\times}(X \| U_d(\mathrm{a}, \mathrm{b})) = -\tfrac{1}{|X|} \sum_{\mathrm{x} \in X} \ln(U_d(\mathrm{a}, \mathrm{b})) = -\tfrac{1}{|X|} \ln(L(X; \mathrm{a}, \mathrm{b})).$$

Consequently, we can minimize cross-entropy by maximizing log-likelihood. This approach allows us to fit optimal parameters in each cluster and minimize the cost function (3).

In the case of uniform distributions the formula for negative log-likelihood function is given as follows

$$H^{\times}(X \| U_d(\mathrm{a}, \mathrm{b})) = -\frac{1}{|X|} \ln(L(X; \mathrm{a}, \mathrm{b})) = -\frac{1}{|X|} \sum_{\mathrm{x} \in X} \ln(U_d(\mathrm{x}, \mathrm{a}, \mathrm{b})) = \ln\left(V_{U_d(\mathrm{x};\mathrm{a},\mathrm{b})}\right).$$

Therefore, our cost function depends on the volume of the support of densities which describes clusters:

$$E(X_1,\ldots,X_k;\mathcal{U}(\mathbb{R}^d)) = \sum_{i=1}^{k} p_i \left(-\ln(p_i) + \ln(V_{U_d(\mathrm{x};\mathrm{a}_i,\mathrm{b}_i)})\right),$$

where $\mathrm{a}_i = \min(X_i), \mathrm{b}_i = \max(X_i)$.

Let us now introduce the algorithm step by step. The UCEC method starts from an initial clustering, which can be obtained randomly or with use of the k-means++ approach.

In our work we use the Hartigan method [20–22]. The aim of Hartigan method is to find partition $X_1,\ldots,X_n$ of X which cost function (3) is as close as possible to the minimum by subsequently reassigning membership of elements from X.

To explain Hartigan approach more precisely we need the notion of group membership function $gr : \{1,\ldots,n\} \to \{0,\ldots,k\}$, which describes the membership of i-th element, where 0 value is a special symbol which denotes that x_i is as yet unassigned. In other words: if $gr(i) = l > 0$, then x_i is a part of the l-th group, and if $gr(i) = 0$ then x_i is unassigned.

Basic idea of Hartigan is relatively simple – we repeatedly go over all elements of X and apply the following steps:

- if the chosen element x_i is unassigned, assign it to the first nonempty group;
- reassign x_i to these group, which decrease cost function;
- check if no group needs to be removed/unassigned, if this is the case unassign its all elements;

until no group membership has been changed.

To implement Hartigan approach for discrete measures we still have to add a condition when we unassign given group. For example in the case of Uniform clustering in R^d to avoid overfitting we cannot consider clusters which contain less then $d+1$ points. In practice while applying Hartigan approach on discrete data we usually removed clusters which contained less then three percent of all data-set.

Observe that in the crucial step in Hartigan approach we compare the cross-entropy after and before the switch, while the switch removes a given set from one cluster and adds it to the other. It means that to apply efficiently the Hartigan approach in clustering it is essential to update parameters.

To calculate cost function we need to calculate minimum and maximum for every dimension of new clusters. If we use simple arrays to keep data, it will take $O(d\cdot k\cdot n^2)$ time per loop (k is number of clusters, d dimension of data). So we use BST tree for every dimension, which gives us min and max in $O(\ln n)$ time. Moreover, we can calculate maximum only after switching point – for change cost function we can just take maximum of current maximums and added point. It enable to decrease time per loop to $O(n \cdot d(\ln n + k))$.

In classical Hartigan approach we switch elements one by one. In the case of uniform distribution this approach is ineffective since the algorithms stacks in local minimums. The effect is caused by the finite support of uniform distributions. In most cases switching one point does not decrease the size of hyperrectangle. Therefore, we consider three possible scenarios of switching points. The classical UCEC switch only

one point. In the second version random UCEC (UCEC-r), we sometimes randomly move some points to another class and minimize it using Hartigan again. It gives better results, but it is time consuming, especially when we apply many random switches.

In the third version multi-points movement UCEC (UCEC-m), we move subsets of points which lie in the borders of supports of uniform densities. The motivation for the solution comes from the observation that for two clusters $X_1, X_2 \subset R^d$ which supports have nonempty intersection, it is profitable to add all points from intersection to the same cluster.

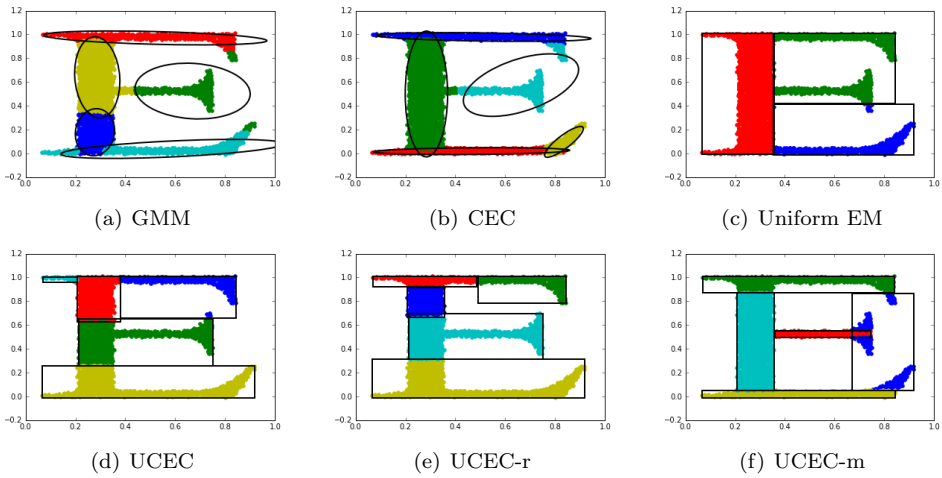
Theorem 2. *Let* $X_1, X_2 \subset \mathbb{R}^d$ *such that* $X_1 \cap X_2 = \emptyset$, $X_1 \cup X_2 = X$ *be given. Let*

$$X_\cap = X \cap supp(U_d(\mathrm{a}_1, \mathrm{b}_1)) \cap supp(U_d(\mathrm{a}_2, \mathrm{b}_2)) \neq \emptyset$$

where $\mathrm{a}_1 = \min(X_1), \mathrm{b}_1 = \max(X_1), \mathrm{a}_2 = \min(X_2), \mathrm{b}_2 = \max(X_2)$ *be such that* $X_\cap \subset X_1$ *and* $E(X_1, X_2, \mathcal{U}(\mathbb{R}^d)) \leq E(X_1 \setminus X_\cap, X_2 \cup X_\cap, \mathcal{U}(\mathbb{R}^d))$. *Then*

$$E(X_1, X_2, \mathcal{U}(\mathbb{R}^d)) \leq E(Y_1, Y_2, \mathcal{U}(\mathbb{R}^d)),$$

for any other clustering such that $\min(X_1) = \min(Y_1), \max(X_1) = \max(Y_1), \min(X_2) = \min(Y_2), \max(X_2) = \max(Y_2)$.



(a) GMM (b) CEC (c) Uniform EM

(d) UCEC (e) UCEC-r (f) UCEC-m

Figure 2. The effect of different clustering algorithms in the case of E-type dataset.

Proof. Cross entropy is equal to:

$$\begin{aligned} E(X_1, X_2, \mathcal{U}(\mathbb{R}^d)) =& p \cdot \left(-\ln(p) + V_{U_d(\mathrm{x};\mathrm{a}_1,\mathrm{b}_1)}\right) + (1-p) \cdot \left(-\ln(1-p) + V_{U_d(\mathrm{x};\mathrm{a}_2,\mathrm{b}_2)}\right) \\ =& p \cdot \ln\left(\frac{V_{U_d(\mathrm{x};\mathrm{a}_1,\mathrm{b}_1)}}{p}\right) + (1-p) \cdot \ln\left(\frac{V_{U_d(\mathrm{x};\mathrm{a}_2,\mathrm{b}_2)}}{1-p}\right). \end{aligned}$$

We consider only such clustering which does not change maximal and minimal values in cluster. Therefore, value of a cost function depends only on p. We can consider a simpler function

$$f(p) = p \cdot \ln\left(\frac{V_1}{p}\right) + (1-p) \cdot \ln\left(\frac{V_2}{1-p}\right).$$

where V_1, V_2 are constant. By analyzing the first derivative of f

$$f'(p) = \ln\left(\frac{V_1}{p}\right) - p \cdot \frac{p}{V_1} \cdot \frac{V_1}{p^2} - \ln\left(\frac{V_2}{1-p}\right) + (1-p) \cdot \frac{1-p}{V_2} \cdot \frac{V_2}{(1-p)^2} =$$
$$= \ln\left(\frac{V_1}{p}\right) - \ln\left(\frac{V_2}{1-p}\right),$$

we obtain that f has one local maximum and no local minimums. Therefore minimum is at one of ends of domain. As a simple corollary we obtain that $E(X_1, X_2, \mathcal{U}(\mathbb{R}^d))$ obtain minimum when all points from $X_\cap$ are in one cluster. □

In natural way it is impossible to verify all possibles subsets which lies in the borders of clusters. But we can take advantage by using Theorem 2 Therefore instead of considering one point we will use all elements which lie in the intersection of supports of considered clusters. k-d trees [23] can be used to increase performance and enable faster search.

Thanks to above modifications and suitable data structures (like k-d trees or BST trees) we obtain effective algorithm for clustering datasets by uniform distributions.

3. Experiments

In this section, we present a comparison of our method with different scenario (UCEC, UCEC-r, UCEC-m) and classical clustering algorithms k-means, GMM, CEC and uniform EM.

In the first example we use letters type datasets (D, E and L), see Fig. 2. To compare the results, we use the standard Bayesian Information Criterion $BIC = ?2LL + k\ln(n)$ and Akaike Information Criterion $AIC = ?2LL + 2k$, 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. We need a number of parameters which are used in our model. The UCEC model uses two scalars for minimal and maximal value for each dimensions $k \cdot 2d$. The results of our experiment are presented in Table 1. In the case of letters which contains uniform distributions on rectangles (letters E and L) our approach (UCEC-m) gives the best results. On the other hand, if data contains curve types structures (letter D) classical approaches fit data with higher precision.

In the second example we use real datasets with labels from UCI repository. In the experiment we use BIC, AIC measures for verify which model fits data best. On the other hand, we use adjusted rand index to check which model is able to recover reference clustering. The results of our experiment one presented in Table 2. Results of recovering clustering for UCEC are comparable with k-means and worse than GMM.

Table 1. The results of classical algorithms in the case of letter-type data.

data	cl.	scoring function	GMM	CEC	U-EM	UCEC	UCEC-r	UCEC-m
D	4	avg l-l	0,479	**0,588**	-0,057	0,487	0,541	0,579
		AIC	-3206	**-3974**	424	-3259	-3625	-3884
		BIC	-3089	**-3943**	540	-3143	-3509	-3768
	5	avg l-l	0,506	**0,736**	-0,070	0,483	0,457	0,606
		AIC	-3377	**-4973**	504	-3224	-3045	-4057
		BIC	-3230	**-4931**	590	-3077	-2898	-3910
	6	avg l-l	0,627	**0,798**	-0,055	0,473	0,600	0,679
		AIC	-4192	**-5387**	378	-3149	-4008	-4545
		BIC	-4014	**-5338**	403	-2972	-3830	-4367
E	4	avg l-l	0,440	0,695	0,284	0,368	0,393	**0,839**
		AIC	-2176	-3487	-1384	-1813	-1941	**-4186**
		BIC	-2065	-3452	-1274	-1702	-1831	**-4074**
	5	avg l-l	0,609	0,829	0,329	0,488	0,534	**0,972**
		AIC	-3019	-4159	-1627	2410	-2589	**-4846**
		BIC	-2880	-4118	-1546	-2270	-2449	**-4706**
	6	avg l-l	0,631	0,829	0,379	0,645	0,677	**1,073**
		AIC	-3117	-4159	-1878	-3191	-3348	**-5345**
		BIC	-2948	-4118	-1797	-3021	-3179	**-5176**
L	4	avg l-l	1,121	1,273	0,506	0,961	1,243	**1,351**
		AIC	-4460	-5097	-2010	-3816	-4950	**-5381**
		BIC	-4353	-5063	-1959	-3710	-4843	**-5274**
	5	avg l-l	1,164	1,315	0,504	1,141	1,255	**1,419**
		AIC	-4623	-5262	-2005	-4528	-4986	**-5644**
		BIC	-4489	-5223	-1955	-4394	-4851	**-5510**
	6	avg l-l	1,160	1,332	0,504	1,077	1,287	**1,432**
		AIC	-4597	-5327	-2005	-4264	-5106	**-5685**
		BIC	-4435	-5283	-1955	-4101	-4944	**-5523**

4. Conclusions

In the paper we construct UCEC, a fast clustering algorithm which describes components by using uniform distributions. In our algorithm we use a data structure like k-d trees or BST trees which allows to implement effective from a numerical optimization point of view, algorithm. Therefore, we obtain a flexible tool for analyzing data with finite support. Moreover, due to its nature UCEC automatically removes unnecessary clusters and therefore can be successfully applied in typical situations where the correct number of groups is not known.

Table 2. The results of classical algorithms in the case of data from UCI repository.

data	scoring function	k-m	GMM	CEC	U-EM	UCEC	UCEC-r	UCEC-m
iris	a-rand	0,730	0,758	**0,901**	0,512	0,772	0,772	0,772
	avg l-l	-	-2,058	**-1,208**	-3,608	-2,270	-2,27	-2,27
	AIC	-	670	**386**	1134	733	733	733
	BIC	-	748	**422**	1213	811	811	811
cancer	a-rand	0,491	**0,755**	0,000	0,068	0,458	0,439	0,545
	avg l-l	-	-3,308	-11,5	-18,70	-3,212	-3,074	**-3,069**
	AIC	-	4006	13193	21520	3898	3740	**3735**
	BIC	-	4532	13454	22046	4423	4266	**4260**
seeds	a-rand	**0,717**	0,679	0,630	0,515	0,632	0,640	0,671
	avg l-l	-	-1,409	**6,079**	-3,080	-1,276	-1,273	-1,311
	AIC	-	680	**-2493**	1382	624	623	639
	BIC	-	827	**-2393**	1529	771	770	786
wine	a-rand	0,371	**0,915**	0,023	0,203	0,571	0,421	0,383
	avg l-l	-	-18,5	**-17,32**	-22,07	-19,97	-19,91	-19,93
	AIC	-	6750	**6351**	8016	7268	7247	7256
	BIC	-	7005	**6644**	8271	7522	7502	7510

5. References

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

[2] Levin M.S., *Combinatorial clustering: Literature review, methods, examples.* Journal of Communications Technology and Electronics, 2015, 60 (12), pp. 1403–1428.

[3] McLachlan G., Krishnan T., *The EM algorithm and extensions.* vol. 382. John Wiley & Sons, 2007.

[4] McLachlan G., Peel D., *Finite mixture models.* John Wiley & Sons, 2004.

[5] Tabor J., Misztal K., *Detection of elliptical shapes via cross-entropy clustering.* In: *Pattern Recognition and Image Analysis.* vol. 7887, Jun 2013, pp. 656–663.

[6] Elguebaly T., Bouguila N., *Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection.* Machine Vision and Applications, 2014, 25 (5), pp. 1145–1162.

[7] Spurek P., *General split gaussian cross–entropy clustering.* Expert Systems with Applications, 2017, 68, pp. 58–68.

[8] Lee S.X., McLachlan G.J., *Finite mixtures of canonical fundamental skew t-distributions.* Statistics and Computing, 2015, pp. 1–17.

[9] Lin T.I., Ho H.J., Lee C.R., *Flexible mixture modelling using the multivariate skew-t-normal distribution.* Statistics and Computing, 2014, 24 (4), pp. 531–546.

[10] Vrbik I., McNicholas P., *Analytic calculations for the em algorithm for multivariate skew-t mixture models.* Statistics & Probability Letters, 2012, 82 (6), pp. 1169–1174.

[11] Browne R.P., McNicholas P.D., *A mixture of generalized hyperbolic distributions.* Canadian Journal of Statistics, 2015.

[12] Śmieja M., Wiercioch M., *Constrained clustering with a complex cluster structure.* Advances in Data Analysis and Classification, pp. 1–26.

[13] Spurek P., Tabor J., Byrski K., *Active function cross-entropy clustering.* Expert Systems with Applications, 2017, 72, pp. 49–66.

[14] Banfield J.D., Raftery A.E., *Model-based gaussian and non-gaussian clustering.* Biometrics, 1993, pp. 803–821.

[15] Jirsa L., Pavelková L., *Estimation of uniform static regression model with abruptly varying parameters.* In: *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on.* 1, IEEE, 2015, pp. 603–607.

[16] Pavelková L., Kárnỳ M., *State and parameter estimation of state-space model with entry-wise correlated uniform noise.* International Journal of Adaptive Control and Signal Processing, 2014, 28 (11), pp. 1189–1205.

[17] Nagy I., Suzdaleva E., Mlynárová, T., *Mixture-based clustering non-gaussian data with fixed bounds.* In: *Proceedings of the IEEE International conference Intelligent systems IS.* 16, 2016, pp. 4–6.

[18] Tabor J., Spurek P., *Cross-entropy clustering.* Pattern Recognition, 2014, 47(9), pp. 3046–3059.

[19] Casella G., Berger R.L., *Statistical inference.* 2. Duxbury Pacific Grove, CA, 2002.

[20] Hartigan, J.A., *Clustering algorithms*, 1975.

[21] Śmieja M., Tabor J., *Spherical wards clustering and generalized voronoi diagrams.* In: *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, IEEE, 2015, pp. 1–10.

[22] Telgarsky M., Vattani A., Hartigan's method: k-means clustering without voronoi. In: *AISTATS*, 2010, pp. 820–827.

[23] Bentley J.L., *Multidimensional binary search trees used for associative searching.* Communications of the ACM, 1975, 18 (9), pp. 509–517.