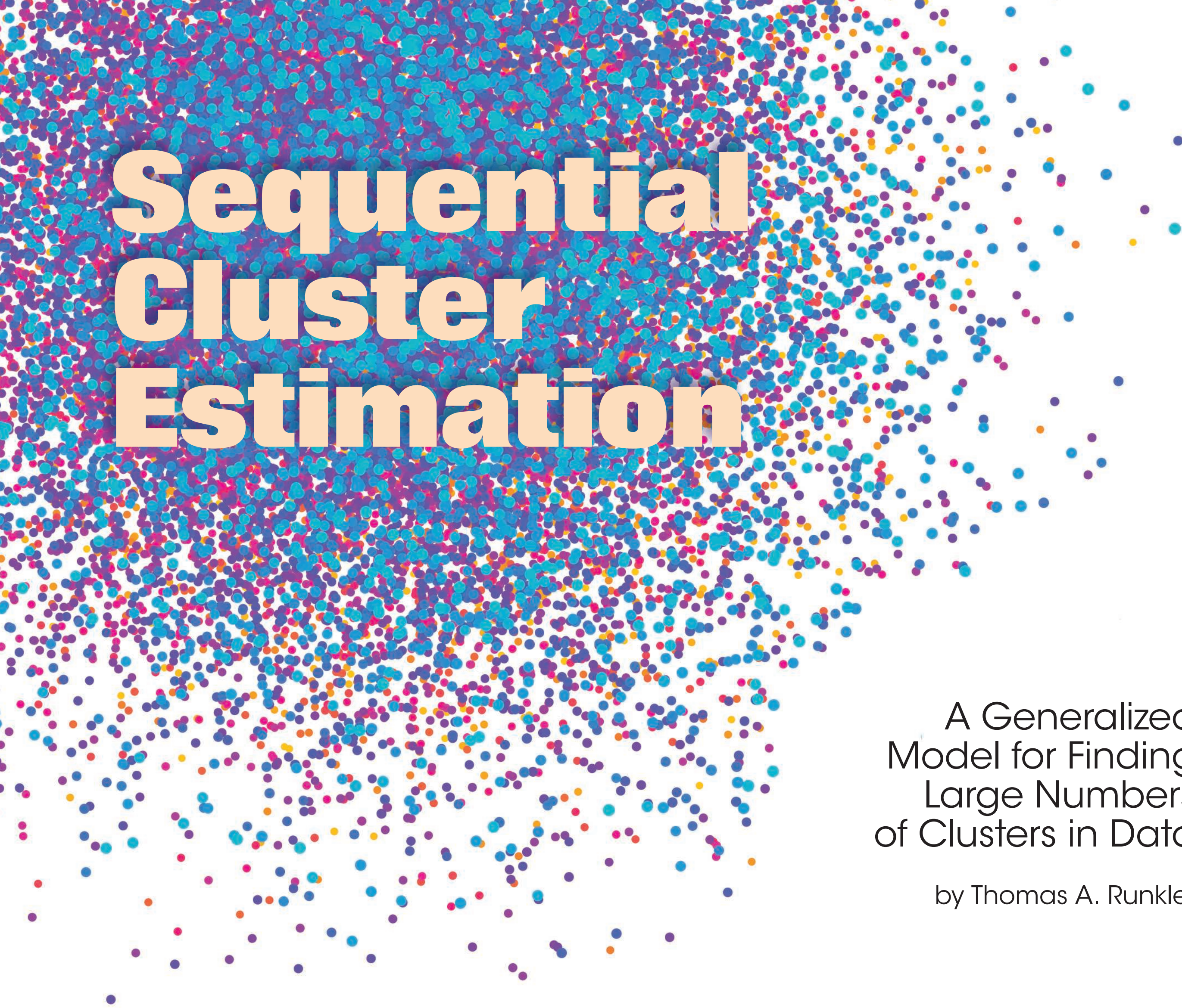

Sequential Cluster Estimation

A Generalized Model for Finding Large Numbers of Clusters in Data

by Thomas A. Runkler

Clustering is an unsupervised learning method that partitions a set of objects into groups (clusters) of similar objects, where similarity is often computed from numerical object feature vectors (also called data points). An early (and still very popular) clustering algorithm, called *k-means* [1], finds clusters by minimizing the sum of the squared distances between data points and associated cluster centers. The k-means algorithm (like many other so-called hard clustering algorithms) assigns each object to one and only one of the considered clusters. In practice, however, cluster assignments may often be ambiguous. Objects may partially belong to several clusters or fit to none of these clusters. Such kinds of ambiguity can be mathematically handled by what is termed *fuzzy set theory* [2]. A fuzzy variant of k-means called *fuzzy c-means* (FCM) has emerged to become one of the most popular fuzzy clustering methods, with hundreds of thousands of scientific publications. For a survey, see [3]–[5]. Fuzzy clustering is often used to generate membership functions for fuzzy rule-based systems [6], [7]. Alternating cluster estimation (ACE) [8] is an extension of FCM for arbitrary membership function shapes. This article introduces sequential cluster estimation (SCE), a variant of ACE that finds clusters sequentially and outperforms nonsequential clustering for data with many clusters.

Digital Object Identifier 10.1109/MSMC.2020.2965319
Date of current version: 20 April 2020

ACE

In cluster partitions generated by FCM, the memberships of each object in all clusters will sum up to one. Even outliers will be partially assigned to the clusters and hence strongly affect the clustering results. This effect is avoided by what is termed *possibilistic c-means* (PCM) [9]. In contrast to FCM, PCM yields convex membership functions, more specifically Cauchy functions, for which outliers will receive close to zero membership values, making the clustering algorithm more robust against outliers. Both FCM

2333-942X/20©2020IEEE

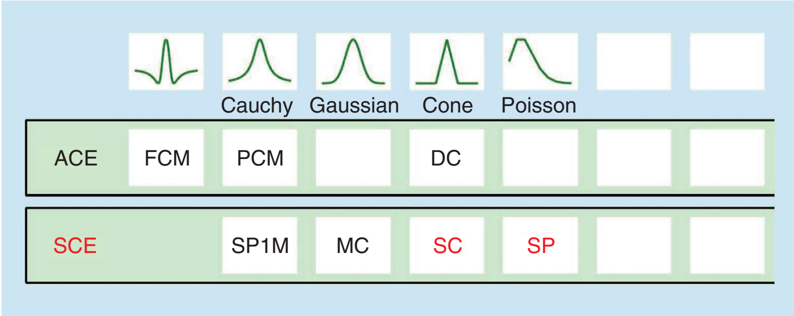

Figure 1. The instances of ACE and SCE for different membership function shapes. New terms introduced in this article are shown in red. DC: dancing cones; SP1M: sequential possibilistic one-means; MC: mountain clustering; SC: sequential cones; SP: sequential Poisson.

and PCM are defined by squared error functionals whose minimization yields specific membership function shapes. Instead, squared error functionals may be abandoned, and fuzzy clustering may be defined directly by membership function shapes (which may or may not correspond to solutions of squared error functionals). This leads to the general scheme of ACE [8], where memberships are updated using the chosen function shape and clusters are updated as the centroid of each cluster, as shown on the left in Table 1. FCM, PCM, and infinitely many other clustering methods are special cases of ACE with specific membership function shapes. Figure 1 shows a taxonomy of ACE (and SCE, as will be discussed later) instances for different membership function shapes such as Cauchy, Gaussian, cone (triangular), Poisson, and many more. The white boxes in the second row of Figure 1 show the ACE instances FCM, PCM (Cauchy membership functions), and dancing cones for conical (i.e., radially triangular) membership functions. Clustering with a conical membership function is illustrated in Figure 2. ACE instances for Gaussian, Poisson, or other membership functions exist but have not obtained specific names (yet). ACE has been successfully applied to a large variety of problems such as function approximation [10], [11], relational clustering [12], data compression [13], web mining [14], keyword extraction [15], or news analysis [16], [17].

SCE

For each update of an FCM cluster estimate, all other clusters are taken into account, so FCM clusters are mutually coupled. In PCM (and many other ACE instances), the clusters are completely independent of each other. This may yield coinciding, almost identical, clusters [18], [19], but it also enables PCM to find only one single cluster, termed *possibilistic one-mean* (P1M) [20], where additional pre- and/or postprocessing is needed to find all desired clusters. One approach to do so is sequential possibilistic one-means (SP1M) [21], where the initial cluster centers are randomly chosen from the given data set—with probabilities proportional to one minus the already assigned memberships—and where the cluster parameters are adapted during the clustering process [22], [23]. An equivalent approach, with Gaussian instead of Cauchy membership functions, is termed *mountain clustering* (MC) [24]. SCE is a generalization of the SP1M method for arbitrarily defined membership function shapes, as listed on the right in Table 1. Notice the differences between the ACE and SCE algorithms on the left and right sides of Table 1, which contain the same commands, but ACE iterates the clusters simultaneously, while SCE finds clusters sequentially, one at a time. The third row of Figure 1 shows the SCE instances SP1M and MC and also two new instances of SCE that will be introduced and experimentally validated in the next sections: sequential cones (SC) and sequential Poisson (SP) clustering. The new contributions of this article (SCE, SC, and SP) are displayed in red.

Finding Many Clusters

With the ability of modern computer systems and cloud services to process large amounts of data, finding large

Table 1. The ACE and SCE algorithms.

ACE	SCE
Initialize cluster centers	For each cluster
Repeat	Initialize cluster center
For each cluster	Repeat
Update memberships	Update memberships
End	Update cluster center
For each cluster	Until termination
Update cluster center	End
End	
Until termination	

numbers of clusters becomes increasingly important, for example, to identify large numbers of different object types in images or large numbers of different customer needs and preferences in customer relation management. The BIRCH data set [25] is an artificial data set that contains an array of 10 × 10 = 100 clusters. Figure 3 shows the clusters found by FCM and by SC clustering. In this example, FCM (left) misses six of the 100 clusters and assigns six duplicate cluster centers, while SC (right) correctly identifies all 100 clusters. These results vary slightly for different initializations (in 1,000 runs, FCM misses an average of 6.6 clusters with standard deviation 1.4), but on average, the SCE approach (here SC) yields a substantially higher cluster coverage than FCM, which indicates that SCE instances are better suited to find many clusters than ACE instances.

In non-black-box machine learning, experts often have substantial prior knowledge about the membership function shapes to be expected.

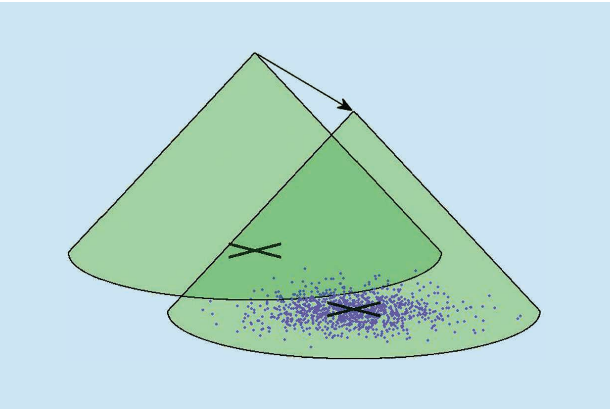

Figure 2. A cone is attracted by a data cluster (blue points) until it matches the local cluster structure.

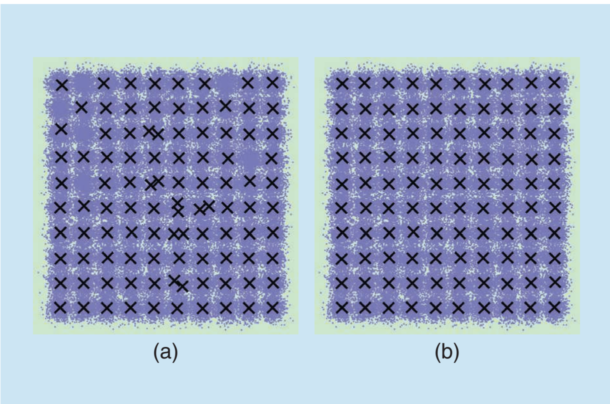

Figure 3. The BIRCH data set clustering results for (a) ACE/FCM and (b) SCE/SC. SCE/SC yields a much better cluster coverage than ACE/FCM.

Clustering Delay Data

In non-black-box machine learning, experts often have substantial prior knowledge about the membership function shapes to be expected. Such knowledge can be easily taken into account in the ACE and SCE clustering schemes. Consider, for example, the case of an automated production line in a factory that makes three different types of products with varying manufacturing times. The production times are not deterministic but vary probabilistically because the necessary components are not immediately available or manufacturing cells or robots may be momentarily busy. Such memoryless waiting times can be mathematically modeled by Poisson distributions. Now consider the problem of identifying the waiting times for the three different manufacturing types from a data set of 500 production pieces (of several varieties). The blue dots in Figure 4 illustrate this for the case of parts with target production times of 30, 40, and 50 s. For this data set, SP clustering, an SCE instance with Poisson membership functions, yields the three green membership functions shown at the top of Figure 4, which precisely match the distributions used to generate this simulation data set (average deviation of the expected value is 2.3%). A similar problem of clustering Poisson distributed data occurs in transcriptome sequencing [26].

Conclusion

This article introduced a new generalized model for sequential clustering called *SCE*. The SP1M and MC methods are instances of SCE with specific membership function shapes (Cauchy and Gaussian). Moreover, two new SCE instances (SC and SP clustering) have been introduced and experimentally validated. For finding large numbers of clusters,

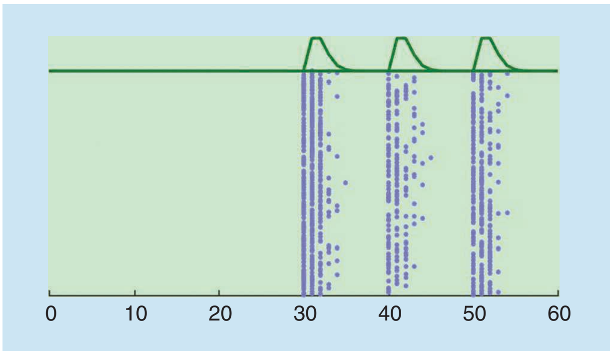

Figure 4. The simulation data from an automated production line with three different product types. SP is able to specifically look for Poisson clusters and correctly identifies the Poisson distributions of the three product specific delays.

SC yields a higher cluster coverage than FCM, which indicates a better suitability for finding many clusters in data. SP is a useful method to find clusters in delay data that can be modeled as mixtures of Poisson distributions.

Due to limited space, many areas of fuzzy clustering research have not been considered here, such as nonpoint prototypes, relational clustering, cluster validity, (visual) tendency assessment, fuzzy rules generation, kernelized clustering, clustering for big data, and many more. The diagram in Figure 1 includes many interesting instances of ACE and SCE, but the remaining white boxes indicate that there is still rich potential for future research, including ACE/SCE instances with other membership function shapes, theory (such as properties and convergence), and further applications.

About the Author

Thomas A. Runkler (thomas.runkler@siemens.com) earned his M.S. and Ph.D. degrees in electrical engineering from the Technical University of Darmstadt, Germany, in 1992 and 1995, respectively, and was a postdoctoral researcher at the University of West Florida, Pensacola, from 1996 to 1997. He has taught computer science at the Technical University of Munich since 1999 and was appointed adjunct professor in 2011. Since 1997, he has worked at the Research and Development Center of Siemens AG in various expert and management functions, currently as a senior principal research scientist. He is a speaker for the Fuzzy Systems and Soft Computing group of the German Association for Computer Science, a member of the IEEE Computational Intelligence Society (CIS) Fuzzy Systems technical committee, and a member of the IEEE CIS Industry Liaison Committee. He is a Senior Member of the IEEE.

References

[1] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Berkeley Symp. Mathematical Statistics and Probability*, 1967, vol. 14, pp. 281–297.

[2] L. A. Zadeh, "Fuzzy sets," *Inf. Control.*, vol. 8, no. 3, pp. 338–353, 1965. doi: 10.1016/S0019-9958(65)90241-X.

[3] M. S. Yang, "A survey of fuzzy clustering," *Math. Comput. Model.*, vol. 18, no. 11, pp. 1–16, 1993. doi: 10.1016/0895-7177(93)90202-A.

[4] A. Baraldi and P. Blonda, "A survey of fuzzy clustering algorithms for pattern recognition," *IEEE Trans. Syst., Man, Cybern., Part B (Cybern.)*, vol. 29, no. 6, pp. 778–801, 1999. doi: 10.1109/3477.809032.

[5] F. Höppner, F. Klawonn, R. Kruse, and T. A. Runkler, *Fuzzy Cluster Analysis: Methods for Image Recognition, Classification, and Data Analysis*. New York: Wiley, 1999.

[6] N. R. Pal, K. Pal, J. C. Bezdek, and T. A. Runkler, "Some issues in system identification using clustering," in *Proc. IEEE Int. Conf. Neural Networks*, Houston, June 1997, pp. 2524–2529. doi: 10.1109/ICNN.1997.614689.

[7] P. Singh, N. R. Pal, S. Verma, and O. P. Vyas, "Fuzzy rule-based approach for software fault prediction," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 5, pp. 826–837, 2016. doi: 10.1109/TSMC.2016.2521840.

[8] T. A. Runkler and J. C. Bezdek, "Alternating cluster estimation: A new tool for clustering and function approximation," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 4, pp. 377–393, Aug. 1999. doi: 10.1109/91.784198.

[9] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, May 1993. doi: 10.1109/91.227387.

[10] T. A. Runkler and J. C. Bezdek, "Function approximation with polynomial membership functions and alternating cluster estimation," *Fuzzy Sets Syst.*, vol. 101, no. 2, pp. 207–218, 1999. doi: 10.1016/S0165-0114(98)00164-X.

[11] T. A. Runkler and J. C. Bezdek, "Regular alternating cluster estimation," in *Proc. European Congr. Intelligent Techniques and Soft Computing*, Aachen, Germany, Sept. 1998, pp. 1355–1359.

[12] T. A. Runkler and J. C. Bezdek, "RACE: Relational alternating cluster estimation and the wedding table problem," in *Proc. Artificial Intelligence Fuzzy–Neuro–Systems '98, München*, W. Brauer, Ed. Mar. 1998, vol. 7, pp. 330–337.

[13] T. A. Runkler, J. Hollatz, H. Furumoto, E. Jünnemann, and K. Villforth, "Compression of industrial process data using fast alternating cluster estimation (FACE)," in *Proc. GI Jahrestagung Informatik (Workshop Data Mining)*, Magdeburg, Sept. 1998, pp. 71–80.

[14] T. A. Runkler and J. C. Bezdek, "Web mining with relational clustering," *Int. J. Approx. Reason.*, vol. 32, nos. 2–3, pp. 217–236, Feb. 2003. doi: 10.1016/S0888-613X(02)00084-1.

[15] T. A. Runkler and J. C. Bezdek, "Automatic keyword extraction with relational clustering and Levenshtein distances," in *Proc. IEEE Int. Conf. Fuzzy Systems*, San Antonio, TX, May 2000, pp. 636–640. doi: 10.1109/FUZZY.2000.839067.

[16] T. A. Runkler and J. C. Bezdek, "Classification of internet newsgroup articles using RACE," in *Proc. Joint IFSA World Congr. and NAFIPS Int. Conf.*, Vancouver, Canada, July 2001, pp. 1437–1442. doi: 10.1109/NAFIPS.2001.943760.

[17] T. A. Runkler and J. C. Bezdek, "Relational clustering for the analysis of internet newsgroups," in *Exploratory Data Analysis in Empirical Research (Studies in Classification, Data Analysis, and Knowledge Organization)*, O. Opitz and M. Schwaiger, Eds. New York: Springer-Verlag, 2002, pp. 291–299.

[18] M. Barni, V. Cappellini, and A. Mecocci, "Comments on 'A possibilistic approach to clustering'," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 393–396, 1996. doi: 10.1109/91.531780.

[19] R. Krishnapuram and J. M. Keller, "The possibilistic C–means algorithm: Insights and recommendations," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 3, pp. 385–393, 1996. doi: 10.1109/91.531779.

[20] J. M. Keller and I. J. Sledge, "A cluster by any other name," in *Proc. Annu. Meeting North American Fuzzy Information Processing Society*, San Diego, CA, 2007, pp. 427–432. doi: 10.1109/NAFIPS.2007.383877.

[21] T. A. Runkler and J. M. Keller, "Sequential possibilistic one-means clustering," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Naples, Italy, July 2017, pp. 1–6. doi: 10.1109/FUZZ-IEEE.2017.8015413.

[22] T. A. Runkler and J. M. Keller, "Sequential possibilistic one-means clustering with variable eta," in *Proc. GMA/GI Workshop Computational Intelligence*. Karlsruhe, Germany: KIT Scientific Publishing, Dec. 2017, pp. 103–116.

[23] W. Wu, J. M. Keller, and T. A. Runkler, "Sequential possibilistic one-means clustering with dynamic eta," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Rio de Janeiro, Brazil, July 2018, pp. 1–8. doi: 10.1109/FUZZ-IEEE.2018.8491499.

[24] R. R. Yager and D. P. Filev, "Approximate clustering via the mountain method," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 8, pp. 1279–1284, 1994. doi: 10.1109/21.299710.

[25] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, 1996. doi: 10.1145/235968.233324.

[26] A. Silva, S. J. Rothstein, P. D. McNicholas, and S. Subedi, "A multivariate Poisson–log normal mixture model for clustering transcriptome sequencing data," *BMC Bioinform.*, vol. 20, no. 1, p. 394, 2019. doi: 10.1186/s12859-019-2916-0.

SMC