JYU DISSERTATIONS 536

Marko Niemelä

Internal Cluster Validation for Data with Missing Values

UNIVERSITY OF JYVÄSKYLÄ
FACULTY OF INFORMATION
TECHNOLOGY

JYU DISSERTATIONS 536

Marko Niemelä

Internal Cluster Validation for Data with Missing Values

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Ylistönrinteen salissa FYS1
kesäkuun 10. päivänä 2022 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in Ylistönrinne, auditorium FYS1, on June 10, 2022 at 12 o'clock noon.

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2022

Editors
Marja-Leena Rantalainen
Faculty of Information Technology, University of Jyväskylä
Ville Korkiakangas
Open Science Centre, University of Jyväskylä

ISBN 978-951-39-9321-4 (PDF)
URN:ISBN:978-951-39-9321-4
ISSN 2489-9003

Permanent link to this publication: http://urn.fi/URN:ISBN:978-951-39-9321-4

ABSTRACT

Niemelä, Marko
Internal Cluster Validation for Data with Missing Values
Jyväskylä: University of Jyväskylä, 2022, 54 p. (+included articles)
(JYU Dissertations
ISSN 2489-9003; 536)
ISBN 978-951-39-9321-4 (PDF)

Clustering is an unsupervised data mining method used to label data into distinct groups. It has numerous applications in various fields, from bioinformatics to object recognition and categorization. The prototype-based clustering methods summarize information in form of cluster centroids that are often called as prototypes. Cluster validation methodology provides a means of assessing the goodness of a clustering solution and identify the optimal number of clusters in the data. Internal cluster validation methods evaluate the quality of clustering by assessing the cluster compactness and separability on the same data set that is input in the clustering phase. A common and sometimes complex issue for both data clustering and cluster validation is the presence of missing values in data that can occur for many different causes, such as non-respondents in questionnaire studies or device operation failures.

This dissertation focuses on extending cluster validation models for treating missing values on data. Since these models are not based on the values of the data vectors but on the computed distances between these vectors, missing value treatment is covered by direct distance estimation between data vectors. The thesis presents a toolbox that is used to demonstrate the usability of the developed methods for research and development purposes. In addition, the background theory of each element of the toolbox and use case examples are proposed. A real-world application is provided where cluster validation is utilized for categorizing learning game players into distinct profiles using a gameplay data in which a part of data values are missing. As the main outcome of the thesis, the missing value handling methods for data preprocessing, clustering, and cluster validation are presented. The functionality and validity of the methods are demonstrated using several numerical experiments and the results confirms the scalability of the techniques and their capability of reliably solving knowledge discovery problems.

Keywords: knowledge discovery, data mining, log data, data preprocessing, missing values, distance computation, distance estimation, clustering, protype-based clustering, number of clusters, cluster validation, internal cluster validation, cluster validation indices

TIIVISTELMÄ (ABSTRACT IN FINNISH)

Niemelä, Marko
Puutteellisen datan klusteroinnin validointi
Jyväskylä: University of Jyväskylä, 2022, 54 s. (+artikkelit)
(JYU Dissertations
ISSN 2489-9003; 536)
ISBN 978-951-39-9321-4 (PDF)

Klusterointi on ohjaamattoman tiedonlouhinnan menetelmä, jota käytetään datan ryhmittelyyn toisistaan poikkeaviin ryhmiin. Klusteroinnilla on lukematon määrä käytännön sovelluksia aina bioinformatiikasta objektien tunnistamiseen ja kategorisointiin. Prototyyppipohjaiset klusterimenetelmät muodostavat annetusta datasta ryhmiä ja kuvaavat tietoa hyödyntäen klusterikeskittymiä, joita kutsutaan myös klusteriprototyypeiksi. Klusterointituloksen sisäistä validointia käytetään mittaamaan klusterirakenteen hyvyyttä hyödyntäen ainoastaan klustereiden muodostamisessa käytettyä dataa. Tavoitteena on pyrkiä löytämään optimaalinen lukumäärä toisistaan erottuvia, tiiviitä joukkoja eli klustereita, jotka kuvaavat dataa parhaiten. Yleinen ongelma klusteroinnissa ja klusterointituloksen validoinnissa ovat puuttuvat arvot. On olemassa useita syitä, joiden vuoksi analysoitavassa datassa esiintyy puuttuvia arvoja. Hyviä esimerkkejä ovat vastaamattomat kysymykset kyselylomakkeissa tai hetkelliset ongelmat laitteistoissa mittausprosessien aikana.

Tässä väitöskirjassa keskitytään laajentamaan klusterivalidoinnin malleja käytettäväksi puuttuvalle datalle. Mallien toiminta perustuu datavektoreiden välillä laskettuihin etäisyyksiin ja tämän vuoksi puuttuvien arvojen käsittely suoritetaan etäisyyksien estimoinnin yhteydessä. Väitöskirjatyössä esitetään avoin ohjelmistokokonaisuus, joka tukee kehitettyjen menetelmien käyttöä tutkimus- ja kehitystoiminnassa. Lisäksi menetelmät kuvataan teoriatasolla ja niiden peruskäyttöä varten tarjotaan useita esimerkkejä. Työssä kuvataan reaalimaailman sovellus, jossa klusterointivalidointia on hyödynnetty ryhmittelemään oppimispelin pelaajia eri profiileihin pohjautuen puuttuvia arvoja sisältävään pelilokidataan. Väitöskirjatyön päätavoitteena oli esittää ja kuvata numeerisilla kokeilla puuttuvien arvojen käsittelymenetelmiä datan esikäsittelyssä, klusteroinnissa ja klusterivalidoinnissa. Tulokset vahvistavat menetelmien skaalautuvuutta kohti luotettavampaa tietämyksen muodostamisprosessia.

Avainsanat: tietämyksen muodostaminen, tiedon louhinta, lokidata, esiprosessointi, puuttuvat arvot, etäisyyksien laskenta, etäisyyksien estimointi, klusterointi, prototyyppipohjainen klusterointi, klustereiden lukumäärä, klusteroinnin validointi, klusteroinnin sisäinen validointi, klusterivalidointi-indeksit

Author

Marko Niemelä
Faculty of Information Technology
University of Jyväskylä
Finland

Supervisors

Professor Tommi Kärkkäinen
Faculty of Information Technology
University of Jyväskylä
Finland

Docent, Ph.D. Sami Äyrämö
Faculty of Information Technology
University of Jyväskylä
Finland

Reviewers

Professor Pauli Miettinen
School of Computing
University of Eastern Finland
Finland

Adjunct Professor Esko Juuso
Faculty of Technology
University of Oulu
Finland

Opponent

Professor Pasi Fränti
School of Computing
University of Eastern Finland
Finland

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and thanks to my supervisor Professor Tommi Kärkkäinen for the patient guidance, encouragement, and advice he has provided throughout my time as a Ph.D. student. He introduced the interesting field of data mining and machine learning to me. He believed me and pushed me to next levels. I am also grateful to my supervisor Docent, Ph.D. Sami Äyrämö whose project was the first where I worked at the University of Jyväskylä. He has scientific expertise in the field of knowledge mining and knows everything about clustering.

I would like to thank my early colleagues Juha-Pekka Kulmala and Jukka-Pekka Kauppi for reviewing my first article, which is not part of this thesis. In addition, I would like to thank my past collaborators in Niilo Mäki Institute: Miia Ronimus, Ulla Richardson, and Heikki Lyytinen. It was a pleasure to work with you. I like to thank my colleague Joonas Hämäläinen for support and many exciting discussions.

I am grateful for the financial support of the Faculty of Information Technology at the University of Jyväskylä and the Niilo Mäki Institute. I would also thank my family members, friends, and relatives for their support outside research work.

Jyväskylä, June 3, 2022

Marko Niemelä

LIST OF ACRONYMS

ACC	Accuracy-index
ADS	Available data strategy
ARI	Adjusted Rand index
CCkNNI	Complete case k-nearest neighbors imputation
CH	Calinski–Harabasz
DB	Davies–Bouldin
DB*	Davies–Bouldin*
CVI	Cluster validation index
EM	Expectation maximization
EED	Expected Euclidean distance
ESD	Expected squared Euclidean distance
GD	Generalized Dunn
ICkNNI	Incomplete case k-nearest neighbors imputation
KCE	kCE-index
KDD	Knowledge discovery in databases
MCAR	Missing completely at random
MAR	Missing at random
MDS	Multidimensional scaling
NMAR	Not missingn at random
NMI	Normalized mutual information
PBM	Pakhira–Bandyopadhyay–Maulik
PDS	Partial distance strategy
RMSE	Root mean square error
RT	Ray–Turi
SIL	Silhouette
SOR	Sequential overrelaxation
WB	WB-index
WG	Wemmert–Gançarski

LIST OF FIGURES

LIST OF TABLES

CONTENTS

LIST OF INCLUDED ARTICLES

PI Marko Niemelä, Tommi Kärkkäinen, Sami Äyrämö, Miia Ronimus, Ulla Richardson, and Heikki Lyytinen. Game learning analytics for understanding reading skills in transparent writing system. *British Journal of Educational Technology, 51(6):2376–2390*, 2020.

PII Marko Niemelä, Sami Äyrämö, and Tommi Kärkkäinen. Comparison of cluster validation indices with missing data. *ESANN 2018, proceedings on European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning, pages 461–466*, 2018.

PIII Marko Niemelä and Tommi Kärkkäinen. Improving clustering and cluster validation with missing data using distance estimation methods. *Computational Sciences and Artificial Intelligence in Industry: New Digital Technologies for Solving Future Societal and Economical Challenges, pages 123–133*, 2022.

PIV Marko Niemelä, Sami Äyrämö, and Tommi Kärkkäinen. Toolbox for distance estimation and cluster validation on data with missing values. *IEEE Access, 10:352–367*, 2021.

Origins of the new methods and implementation

In Articles PI, PII, PIII, and PIV ideas originated from Tommi Kärkkäinen and the author implemented the methods. In Articles PIII and PIV, the implementation of K-spatialmedians clustering are from the early implementation created by Sami Äyrämö and Tommi Kärkkäinen. In Articles PIII and PIV, the available data strategy method was based on the early work of Tommi Kärkkäinen. In Article PIV, the implementation of the data set generator for multidimensional data originated from the work of Ph.D. Joonas Hämäläinen.

Writing the articles

In Articles PI, PII, and PIV, the author's contribution to writing each article was significant. In Article PIII, the authors' contribution to writing were close to equal.

1 INTRODUCTION

Caused by multiple types of powerful sensors, advanced digitalization techniques, and significantly increased storage capabilities, big data in the sense of the size of data sets, high dimensionality data, speed of data accumulation, heterogeneous data format, or data quality create one of the significant challenges facing machine learning today [54]. It was estimated that the digital universe would consume approximately 44 billion terabytes (zettabytes) at the end of 2021 [35]. Most of the stored data is in electronic media, which have high potential for developing automatic data analysis and retrieval techniques. In addition to increased data volumes, the data availability in various forms (e.g., text, image, and video) has also increased. Nowadays, mobile phones with video cameras and internet connections have spurred a massive amount of internet traffic, images, and videos. Millions of low-cost sensors measure a broad range of information from the environment and transmit data regularly. There are many domains to obtain data, e.g., telecommunication, internet search, social network, finance, health care, etc.

In general, the databases are increasing in two ways: the number of observations in the database and the number of variables in each observation. Manual data analysis is becoming slow, expensive, and utterly impractical in many domains as data volumes grow dramatically. The knowledge discovery in databases (KDD) process consists of mapping low-level data, which is typically too voluminous to understand and digest easily, into other forms that might be more compact, abstract, or useful [29]. The KDD process focuses on the overall process of knowledge discovery from data sets, including how the data is stored and accessed, how methods scale to enormous volumes of data and still run efficiently, how missing values and noisy data are handled, how resulting models can be interpreted as useful or interesting knowledge, and how human-machine interaction can be supported.

The KDD process is illustrated in Figure 1. The process starts in the form of understanding the application domain and identifying the potential goals. The target data is selected by focusing on a subset of variables on which discovery is performed. The data preprocessing is applied, which possibly includes removing noise, scaling variables to the same range, and deciding strategies for handling

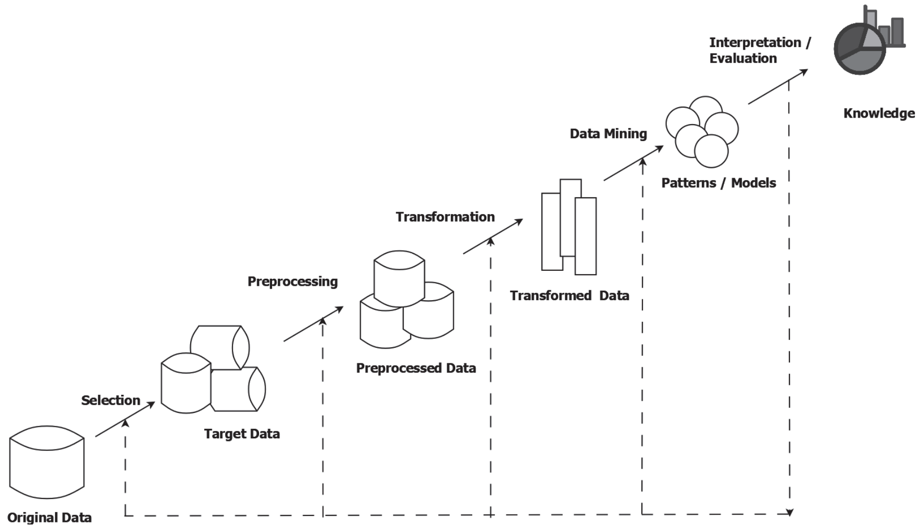


FIGURE 1 Steps of the KDD process

missing values. In the data transformation step, valuable features are selected with appropriate methods (e.g., using feature selection [8]), or data dimension is reduced via a dimension reduction technique [46, 63]. Then, a particular data mining model is selected and utilized, e.g., classification, regression, or clustering model. The step includes tuning the essential parameters of the chosen model. After that, the results will be interpreted or evaluated. There exists a possibility to return previous actions for further iteration(s). Finally, the obtained knowledge is discovered and checked against conflicts with previous beliefs.

Clustering is an essential component of various data analysis or machine learning applications (e.g., regression, prediction, and data mining [37]). The primary purposes of clustering are to get data insights, generate hypotheses, detect anomalies, identify salient features, identify the degree of similarity among data vectors, and organize and summarize data through cluster prototypes [43]. For instance, collecting and labeling a large set of observations can be surprisingly costly. One might wish to train large amounts of unlabeled data, and then use supervision to label the groupings found. Further, unsupervised methods can be used for finding features that will be useful for categorization. This procedure can be called smart preprocessing or smart feature extraction [38].

Clustering has numerous amount of applications from diverse fields. For instance, applications can be related to bioinformatics, character recognition, information retrieval, image clustering, object recognition, learning analytics, etc. In [91], the application of clustering algorithms in bioinformatics was described. The central assumption was that functionally similar genes or proteins usually share similar patterns or primary sequence structures. In character recognition [15], clustering was used to identify lexemes in handwriting text for writer-independent handwriting recognition. Information retrieval is concerned with the automatic storage and retrieval of documents [74]. For instance, libraries can use information retrieval systems to provide access to books, journals, and other documents. A widespread clustering application is image clustering [33, 39]. Image clustering of its colors is often referred to as image signal quantization or image compression, because it leads to a reduced number of colors and image data. Image quantization can be used as a preprocessing method, e.g., in image database mining or content-based image retrieval. Reduced size and simplified color structure make knowledge mining and object searching from large image databases easier as the most important information remains in the quantized image. In [23], the use of clustering to group views of 3D objects for object recognition was described. The system employed a viewpoint-dependent approach to the object recognition problem. Each object to be recognized was presented in an image library consisting of images of that object.

Learning analytics is defined as "the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs" [16]. The common data sources are learning management systems, questionnaires, student information systems, and learning games. Clustering methods are commonly used in learning analytics. For example, in [90], the learners were linked via clustering for those with different individual and social behavior patterns. The results consisted of different cluster profiles characterizing learners who were personally participative but less communicative, collaboratively participating but shallow learners, and less participative poor learners. In [55], the clustering model was constructed to give valuable information on algebra-solving skills components. In [6], multiple clustering methods at various stages of data analysis were utilized to identify different patterns of the development of programming behavior in an undergraduate programming course. In [89], clustering was used to analyze the problem-solving patterns of learners for open-ended engineering tasks. The features related to learners' actions were extracted from hand-coded video data. The results revealed that designed engineering practices were closely related to learners' experience level.

In PI, the learning analytics application for analyzing learners in a learning game called *GraphoLearn* was proposed. The learners' selections of letter-sound tasks were gathered from 1632 players who were 6.5–8.75 years old. Clustering and cluster validation were used to identify distinct player profiles. Validation indices identified six different learner profiles for players who used lowercase letters (1275 players). The erroneous selections and calculated statistics offered valuable information about the cluster profiles. This information can be used, for example, as support for tracking children with certain types of bottlenecks compromising reading skills and tailoring the learning environment for individual needs.

When preparing data for modeling, several problems need to be addressed. One of these is missing values in data. Many, if not most, modeling tools have difficulty digesting missing values. For instance, standard clustering methods like K-means clustering, can not be used for cluster analysis if missing values are not ignored or imputed. In most situations, simple techniques (e.g., overall mean imputation) for handling missing data lead to inefficient analyses and, more seriously, severely biased data models. During past decades, several preprocessing methods have been developed for data sets with missing values.

2 DATA PREPARATION

In [72], it was suggested that more than half of the total time required to complete a data mining project should be spent on data preparation since it is one of the most vital parts of the project's success. Selected and preprocessed data significantly impacts the final models and, therefore, the quality of the knowledge. At the same time, the modified data can either facilitate or complicate further the KDD process. Hence, the data preparation must be done with care. The main problems with real-world data are noise, missing values, and inconsistent data due to mechanical or human errors. High-dimensional irrelevant data is also one of the challenges.

Data characterization describes data in a meaningful way. In [26], data were represented using: the number of classes, observations, attributes, and features. In addition, parameters of location and dispersion can be measured to describe the data set. Location parameters include measurements such as minimum, maximum, arithmetic mean, and median. Dispersion parameters include the range and standard deviation of a single feature. The parameters which can deal with extreme values or outliers are called robust parameters.

Data visualization before preprocessing can help understand data to identify missing values and outliers, and identify relationships among attributes. Further, visualization can help in finding appropriate preprocessing methods for data.

Data cleaning, data reduction, data transformation, and data integration are common preprocessing methods in data mining [72]. Data cleaning consists of imputing of missing values, smoothing noisy data with random errors, removing outliers, and resolving inconsistent data. Data reduction aims to reduce the volume of data and produce similar analytical results by removing repeated observations, applying dimension reduction techniques to remove irrelevant and redundant attributes, or discretizing continuous valued data. Data compression uses encoding mechanisms to obtain a reduced representation of the original data. Through data transformation, text or graphical data can be converted a format that can be further processed. In addition, the transformation includes scaling or normalization of numerical data. The method is essential in distance-based applications since distance measurements taken by large-scale attributes outweigh small-scale attributes. Data generalization is one part of transformation. The goal is to replace data with higher lever concepts to aggregate the KDD process by combining multiple attributes with the same categories to one attribute with that category. Data integration combines data from several sources and corrects differences in coding schemes. Attributes representing a given concept may have different names in different databases. Therefore, extra care must be taken to avoid inconsistencies and redundancies of data.

2.1 Data scaling

Data scaling or feature scaling is utilized during the data preprocessing step. A method is used to normalize the range of independent variables of data. Hence, the weights of different variables are equalized after scaling process. The scaling is necessary for distance-based applications because if one of the variables dominates, i.e., has a broad range of values, the distance is mainly governed by this particular variable. Typically z-score or min-max normalization is used [66]. The z-score transforms variables to zero mean and unit variance, which can be realized as follows:

$$x' = \frac{x-\mu}{\sigma} = \frac{1}{\sigma}x - \frac{\mu}{\sigma} = \alpha x - \beta, \tag{1}$$

where x is the original variable, x' is the scaled variable, μ is the sample mean, and σ is the standard deviation. One can see that this is a linear transformation of the variable. By determining the coefficients differently, other approaches, such as min-max scaling to a specific range, are obtained. The selection of the target range depends on the nature of the data. The most common choices are $[-1, 1]$ and $[0, 1]$. To rescale a range between an arbitrary set of values $[a, b]$, $a < b$, a, $b \in \mathbb{R}^1$, the formula read as:

$$x' = a + \frac{(x - \min(x))(b-a)}{\max(x) - \min(x)} = \frac{(b-a)}{\max(x)-\min(x)}x + \frac{a\max(x) - b\min(x)}{\max(x)-\min(x)}. \tag{2}$$

Hence, if we select the range of $[0, 1]$ the coefficients becomes:

$$\alpha = \frac{1}{\max(x)-\min(x)} \text{ and } \beta = -\frac{\min(x)}{\max(x)-\min(x)}.$$

The potential problem in variable scaling methods is the existence of outliers. For instance, in the range scaling approach, one outlier forces the rest of the data to the other end of the range. It is also good to notice that, for the scaled data set, binary variables contribute to the distance computation with the maximum influence.

2.2 Missing values

The performance of clustering for incomplete data depends on the types of missing values in a data set. The missing data refers to the relationship between missigness and the underlying variable values in the data set [57]. There are three types of missing data approaches. The missing values are called missing completely at random (MCAR) if variable being missing is independent both observed and unobserved variables. The missing values are denoted as missing at random (MAR) when the missingness is not random, but where missingness can be accounted by observed variables. If the missingness depends on the missing values in the data set, the approach is called not missing at random (NMAR). Even though, MCAR represents the general case of missingness, MAR and MCAR are standard approaches in practice. For example, on questionnaires the question for socioeconomic status often remains unanswered by young people; thus, the socioeconomic status remains unanswered depending on age, and the missing values in that feature are MAR. Further, high-income earners do not often answer the question. Hence, missing values in the feature socioeconomic status are NMAR because values depend on themselves. In many cases, missing data approach is not known beforehand, but it can be validated with statistical testing [58].

2.2.1 Imputation

In statistics, the approach to filling missing values with substituted values is called imputation. When complete observation is substituted, it is known as unit imputation, whereas substituting a variable of an observation is known as item imputation. The missing values can introduce a high amount of bias, making analysis of data more challenging and creating reduction in efficiency. Imputation solves the problem related to listwise deletion by replacing missing data based on the estimated value of other available information. There are several methods for imputing missing values with machine learning methods [57]. Conditional mean imputation uses estimators to predict the incomplete observations in the data set. The method is optimal in terms of the mean squarred errors of the imputed values but suffers biased derived statistics of data. For example, the variance of data set is not consistently estimated. Randomly drawn imputation substitutes values from different underlying distributions. However, the method has too much variability in estimates of any single value to be sufficiently accurate. Multiple imputations do not create a single but several or multiple imputed data sets in which different imputations are based on a random draw from different estimated underlying distributions. Each imputed data set can be analyzed using standard analytical techniques. The estimates can be averaged to get a pooled estimate of the associations. The mean of standard errors is a measure of uncertainly in the estimated underlying distributions of the observations with missing values. However, repeating the analysis several times can be impractical as training and analyzing a sophisticated model usually tends to be computationally intensive.

A low-rank matrix completion method is an alternative way to impute missing values of a data set. The low rank matrix has a decreased number of freedom and, therefore, it makes the estimation problem of missing values practical to solve. The rank minimization can be addressed by using convex relaxation techniques utilizing the nuclear norm. The algorithm substitutes missing values iteratively until the final convergence is reached, corresponding approximately to the best estimates of the missing values [62, 61].

In [87] an extension of the complete case *k*-nearest neighbors imputation (CCkNNI) method called incomplete case *k*-nearest neighbors imputation (ICkNNI) was presented. As the name suggests, the CCkNNI method imputes missing values from *k*-nearest's complete observations. This restriction has severe problems, especially when the amount of missing values is high or when there are few or no complete observations. An alternative version (the ICkNNI method) overcomes the restriction in presenting the case library, allowing some incomplete observations as alternatives for imputation. The case library consists of the set of observations that depend on the observation value being imputed. The eligible nearest neighbors are those which have the same subset of observed values as imputed observation and the imputed value is available in the subset. The missing value is imputed by the sample mean if there are insufficient neighbors.

All imputation mechanisms produce some bias to the estimates of substituted values. However, if the fraction of missing values is sufficiently small, it is reasonable to select some imputation methods for filling missing values and proceed with further processing. Errors related to inaccurate imputation may be considered insignificant to the results through the whole KDD processing pipeline. With a larger proportion of missing values, errors caused by the imputation can be high. For example, in [28] an analysis on the effect of imputation on classification error for discrete data was proposed.

2.2.2 Expectation maximization

The expectation maximization (EM) is an iterative relocation algorithm, which estimates the unknown parameters of statistical models when the models involve latent variables in addition to known observations and unknown parameters. The method is covered with a range of different distributions and used in cases where unknown parameters cannot be solved directly. The missing values occur in the data set or the model assumes the existence of unobserved latent variables. For example, a mixture density model can be described by assuming each observation has corresponding latent variable that specifies the mixtute component to which each observation belongs to [20, 5]. Typically, solving the equation for parameters of distribution requires the values of latent variables and vice versa. One can simply initialize one of the two sets of unknowns with arbitrary values to get the first estimate of the second set and use these estimates to get better estimates for the first set. One can keep alternating between these two steps until the resulting values converge into fixed numbers. The EM method can identify local optimums, but the global optimum is not guaranteed. The EM alternates between two steps: expectation and maximization. The expectation step creates the function of expected values for the latent variables using the current estimates for the parameters. The maximization computes parameters that maximizes the likelihood function found during the expectation step. The new estimates are then used to determine the distribution of the latent variables in the next iteration. The EM method is commonly used in the treatment of missing data (see, e.g., [25, 81, 70, 68]) but also in learning Gaussian mixture models for labeling data to different clusters (see, e.g., [20, 19, 5, 4]).

2.2.3 Distance computation strategies

In most pattern recognition applications, distances between data points is the main interest. The commonly used distance measure is the Euclidean distance. For all complete elements of both data vectors, this is a straightforward arithmetic to compute. But if one or more of the elements are missing, the distance between the vectors is not obvious.

Distance-based computation methods (e.g., traditional K-means clustering) can be adapted for strategies, which can handle missing values. The partial distance strategy (PDS) is already used in the context of k-nearest neighbors search [22]. The original version of PDS is given in [83]. The partial-based l_2-distance between two data vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ in a n-dimensional space can be presented as follows:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\frac{n}{n^*} \sum_{i=1}^{n} ((\mathbf{p}_1)_i (\mathbf{x}_1)_i - (\mathbf{p}_2)_i (\mathbf{x}_2)_i)^2}, \tag{3}$$

where n^* denotes pairwise-known values and

$$(\mathbf{p}_k)_i = \begin{cases} 1, & \text{if } (\mathbf{x}_k)_i \text{ exists,} \\ 0, & \text{otherwise.} \end{cases}$$

The PDS method can be extended to other norms. For example, the partial-based l_1-distance is defined as:

$$d(\mathbf{x}_1, \mathbf{x}_2)_1 = \frac{n}{n^*} \sum_{i=1}^{n} |(\mathbf{p}_1)_i (\mathbf{x}_1)_i - (\mathbf{p}_2)_i (\mathbf{x}_2)_i)| \tag{4}$$

The scaling with the term (n/n^*) is performed during the computation. The method without scaling ($n^* = n$) is called an available data strategy (ADS). The ADS has a long record and it was first presented in the context of building a computationally efficient multigrid system for representing boundary value problems [51]. Later on, the ADS method is used in K-spatialmedians clustering [77, 52].

2.2.4 Distance estimation strategies

Imputations and distance computations strategies are alternative ways to handle missing values. However, the methods can lead to a suboptimal estimate of the distance. There are also methods to estimate all pairwise distances directly in a data set, which enables the use of machine learning techniques without having to consider any further tricks to deal with the missing values [25, 64].

The expected squared Euclidean distance (ESD) between two data vectors can be divided into four parts depending missing and observed values of each observation:

$$\begin{aligned} E\left[\|\mathbf{x}_i - \mathbf{x}_j\|^2\right] &= \sum_{l \in A_i \cap A_j} \left((\mathbf{x}_i)_l - (\mathbf{x}_j)_l\right)^2 + \sum_{l \in A_i \cap M_j} E[((\mathbf{x}_i)_l - (X_j)_l)^2] \\ &+ \sum_{l \in M_i \cap A_j} E[((X_i)_l - (\mathbf{x}_j)_l)^2] + \sum_{l \in M_i \cap M_j} E[((X_i)_l - (X_j)_l)^2], \end{aligned}$$

where A_i and A_j denote the available values of data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, and M_i and M_j denote the missing values of the vectors. Note that the missing value can be replaced with a random variable denoted by $(X_i)_l$ for every $l \in M_i$. The equation can be expanded as follows:

$$\begin{aligned} E\left[\|\mathbf{x}_i - \mathbf{x}_j\|^2\right] = &\sum_{l \in A_i \cap A_j} \left((\mathbf{x}_i)_l - (\mathbf{x}_j)_l\right)^2 \\ &+ \sum_{l \in A_i \cap M_j} \Big(((\mathbf{x}_i)_l - E[(X_j)_l])^2 + \mathrm{Var}[(X_j)_l]\Big) \\ &+ \sum_{l \in M_i \cap A_j} \Big((E[(X_i)_l] - (\mathbf{x}_j)_l)^2 + \mathrm{Var}[(X_i)_l]\Big) \\ &+ \sum_{l \in M_i \cap M_j} \Big((E[(X_i)_l] - E[(X_j)_l])^2 + \mathrm{Var}[(X_i)_l] + \mathrm{Var}[(X_j)_l]\Big). \end{aligned}$$

In more detail, the second summation can be written:

$$\begin{aligned} E\left[((\mathbf{x}_i)_l - (X_j)_l)^2\right] &= E\left[(\mathbf{x}_i)_l^2 - 2(\mathbf{x}_i)_l(X_j)_l + (X_j)_l^2\right] \\ &= (\mathbf{x}_i)_l^2 - 2(\mathbf{x}_i)_l E\left[(X_j)_l\right] + E\left[(X_j)_l^2\right] \\ &- E\left[(X_j)_l\right]^2 + E\left[(X_j)_l\right]^2 \\ &= ((\mathbf{x}_i)_l - E\left[(X_j)_l\right])^2 + E\left[(X_j)_l^2 - E\left[(X_j)_l\right]^2\right] \\ &= ((\mathbf{x}_i)_l - E\left[(X_j)_l\right])^2 + \mathrm{Var}\left[(X_j)_l\right] \end{aligned}$$

Let us assume a Gaussian distributed data and random variables are MAR. The estimates for missing values can obtained using mean and covariances of conditional multivariate Gaussian distribution by conditioning missing values with observed ones [25]. The final form of the equation is:

$$E\left[\|\mathbf{x}_1 - \mathbf{x}_2\|^2\right] = \sum_{i=1}^{n} \left(((\mathbf{x}'_1)_i - (\mathbf{x}'_2)_i)^2 + (\boldsymbol{\sigma}'_1)_i^2 + (\boldsymbol{\sigma}'_2)_i^2\right), \tag{5}$$

where

$$(\mathbf{x}'_k)_i = \begin{cases} (\mathbf{x}_k)_i, & \text{if } i \in A_k; \\ E[(\mathbf{x}_k)_i \mid (\mathbf{x}_k)_{A_k}], & \text{if } i \in M_k; \end{cases}$$

$$(\sigma'_k)^2_i = \begin{cases} 0, & \text{if } i \in A_k; \\ \mathrm{Var}[(\mathbf{x}_k)_i \mid (\mathbf{x}_k)_{A_k}], & \text{if } i \in M_k. \end{cases}$$

The kth observation is a normally distributed with the mean vector and the covariance matrix:

$$\begin{aligned} (\boldsymbol{\mu}_{\boldsymbol{k}}')_{M_k} &= (\boldsymbol{\mu})_{M_k} + \Sigma_{M_k A_k}\Sigma^{-1}_{A_k A_k}\left((\mathbf{x}_k)_{A_k} - (\boldsymbol{\mu})_{A_k}\right), \\ \Sigma'_{M_k M_k} &= \Sigma_{M_k M_k} - \Sigma_{M_k A_k}\Sigma^{-1}_{A_k A_k}\Sigma_{A_k M_k}. \end{aligned} \tag{6}$$

Missing values of $\mathbf{x}'_k$ can be imputed from $(\boldsymbol{\mu}_{\boldsymbol{k}}')_{M_k}$ vector and $\boldsymbol{\sigma}'^2_k$ is a sum of diagonal elements of $\Sigma'_{M_k M_k}$ matrix.

Estimating $\boldsymbol{\mu}$ and Σ for incomplete data is not a trivial task. These parameters can be updated iteratively using the EM algorithm with the maximum negative log-likelihood convergence criterion [25]. However, the convergence is not guaranteed if the number of missing values is high compared to available values.

In [64], the idea of the ESD was extended to the expected Euclidean distance (EED). The EED requires the same assumption of Gaussian distributed data as the ESD. In addition, it is assumed that the squared Euclidean distances follow a Gamma distribution. It is reasonable to choose the Nakagami distribution [69] instead of the Gamma since a random variable from Nakagami can be obtained from the square root of Gamma's distributed value. The Nakagami distribution is a function of shape (m) and spread (Ω) parameters. Hence, a random variable of the EED can be computed as follows:

$$\begin{gathered} E\left[\left(\sum_{i=1}^{n}((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^2\right)^{\frac{1}{2}}\right] = E\left[z^{\frac{1}{2}}\right] = \frac{\Gamma(m+\frac{1}{2})}{\Gamma(m)}\left(\frac{\Omega}{m}\right)^{\frac{1}{2}}, \\ m = \frac{E\left[z\right]^2}{\mathrm{Var}\left[z\right]}, \ \Omega = E\left[z\right], \end{gathered} \tag{7}$$

where Γ is the Gamma function. The variance can be expressed as

$$\begin{aligned} \mathrm{Var}[z] &= \mathrm{Var}\left[\sum_{i=1}^{n}((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^2\right] \\ &= \sum_{i=1}^{n}\mathrm{Var}\left[((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^2\right] \\ &= \sum_{i=1}^{n}E\left[((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^4\right] - E\left[((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^2\right]^2 \\ &= \left(\sum_{i=1}^{n}E\left[(\mathbf{x}_1)^4_i + (\mathbf{x}_2)^4_i - 4(\mathbf{x}_1)^3_i(\mathbf{x}_2)_i - 4(\mathbf{x}_1)_i(\mathbf{x}_2)^3_i + 6(\mathbf{x}_1)^2_i(\mathbf{x}_2)^2_i\right]\right) \\ &\quad - \sum_{i=1}^{n}E\left[((\mathbf{x}_1)_i - (\mathbf{x}_2)_i)^2\right]^2, \end{aligned}$$

where the expected values are obtainable using the non-central moments of normal distribution:

$$\begin{aligned}
E[\mathbf{x}_k] &= \hat{\mathbf{x}}_k \\
E[\mathbf{x}_k^2] &= \hat{\mathbf{x}}_k^2 + \boldsymbol{\sigma}_k^2 \\
E[\mathbf{x}_k^3] &= \hat{\mathbf{x}}_k^3 + 3\hat{\mathbf{x}}_k\boldsymbol{\sigma}_k^2 \\
E[\mathbf{x}_k^4] &= \hat{\mathbf{x}}_k^4 + 6\hat{\mathbf{x}}_k^2\boldsymbol{\sigma}_k^2 + 3\boldsymbol{\sigma}_k^4.
\end{aligned}$$

Note that weighted moments are needed if the data are assumed to follow the Gaussian mixture distribution (see e.g., [64]).

3 CLUSTERING

In clustering, a set of observations is decomposed into groups ("clusters") such that the pairwise dissimilarities between those assigned to the same cluster tend to be smaller than those in different clusters [39]. Cluster analysis is the formal study of algorithms and methods for grouping or classifying observations by their measured or perceived characteristics or similarities [43]. The absence of category information, labels, distinguishes unsupervised clustering from supervised classification. The objective of cluster analysis is simply to find a convenient and valid stutructure of the data and not to establish rules for separating future data into distinct categories. Hence, clustering is explanative, predictive, and descriptive in nature. In addition, it investigates multivariate data sets that contain different data types. In general, clustering is more challenging compared to labeled classification. The measures of similarity and the evaluation criterion are main components of clustering. The most common approach for defining similarity is measuring the distance among the data patterns like squared Euclidean distance.

The similarity of observations within a cluster has a significant role in the clustering process. The clusters can differ in terms of their shape, size, variance, density, and the presence of noise which makes the detection of the cluster even more complex. However, clusters are ideally compact and isolated in a data space. The similarity of a cluster is mainly measured through the summed distance between cluster centroid and the observations within a cluster. The valid distance measure should be symmetric, i.e., $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$, and obtain minimum value (ideally zero) in case of equal vectors [76]. In addition, the distance measure is called a metric distance measure if it satisfies the following triangle inequality (it is assumed that vectors are complete):

$$\begin{aligned} &d(\mathbf{x}_i, \mathbf{x}_k) \leq d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_k) \quad \forall\ \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathbb{R}^n, \\ &d(\mathbf{x}_i, \mathbf{x}_j) = 0 \Rightarrow \mathbf{x}_i = \mathbf{x}_j \quad \forall\ \mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^n. \end{aligned} \tag{8}$$

There are different methods for clustering: hierarchical [67], partitional [53], fuzzy [24], grid based [37], graph based [93], density based [91, 92], and model based [73, 40, 31]. However, in general, clustering algorithm can be divided into two main groups: hierarchical and partitional [43]. Hierarchical clustering recursively identifies a nested sequence of partitions and visualizes the result as a dendogram, which enables to see how observations are being merged into clusters or split successive levels of similarity. Hierarchical clustering proceeds either from bottom to up in agglomerative mode, starting with each observation in its own cluster and merging most similar pair of clusters successively to form a cluster hierarchy, or from up to bottom in divise mode, starting with all observations in one cluster and recursively dividing each cluster into smaller clusters. One can then select a clustering at some fixed level of similarity, which makes the most sense for the given application. Partitional clustering algorithms simultaneously partition of the data, attempting to recover natural groups present in the data. Partitional clustering methods have advantages in applications that consist large data sets for which constructing a dendogram is computationally expensive.

3.1 Partitional clustering

Partitional clustering attempts to determine the K partition of N observations in n-dimensional space such that the observations in a cluster are more similar to each other than the observations in different clusters. Solving the problem requires determining the clustering error criterion (e.g., sum of the squared errors). The local criterion forms clusters by applying the local structure of data. For example, identifying high-density regions in the data space. The global clustering criterion represents each cluster by a prototype and assigns the observations to clusters according to the most similar prototypes [44].

The theoretical solution to this partitional problem is straightforward. Simply select a criterion, evaluate it for all possible partitions containing K clusters, and pick the partition that optimizes the criterion. However, the task may not be easy. First, the mathematical formula for transforming data partitions into so called "clusters" may not be obvious. The formula is required to be simple enough for computational reasons, but relatively complex to reflect anomalous data structures. Only a small number of clustering criteria can be understood both mathematically and intuitively. The single best clustering error criterion does not exist, because the notion of cluster depends on the application, and it is usually weakly defined [82, 76, 12]. It is also good to rememeber that clusters are, in large part, on the eye of the beholder [27].

Secondly, the number of K various partitions for data is astronomical, even for small numbers of observations, and evaluating even the simplest criterion can be impractical. More specifically, grouping N observations into K groups forms the equation, which is solved by Stirling numbers of the second kind [13]:

$$S(N,K) = \frac{1}{K!}\sum_{i=1}^{K}(-1)^{K-i}\binom{K}{i}(i)^N \,, \tag{9}$$

which can be approximated by $K^N/K!$. For example, an exhaustive search for the best set of $K = 5$ clusters in $N = 100$ observations would require computing more than 10^{67} partitions, so most exhaustive searches are, therefore, infeasible.

A more practical approach than exhaustive search is iterative optimization. The basic structure of iterative relocation methods is given in Algorithm 1. The clustering via distribution-based models is possible through the EM method (as described in Section 2.2.2). The expectation step computes the expectation of the posterior of the latent variables (cluster labels). The maximization step optimizes the model parameters (typically centroid locations and covariances) to fit the data best. The steps are repeated iteratively until the convergence criterion (e.g., not significant change in likelihood functions) is obtained.

The prototype-based clustering methods use centroids of clusters to represent the prototypes. The mean and median are common choices for the estimates. The algorithm does not necessarily find the global optimum of the clustering error function, but the local optimum is guaranteed. Therefore, Algorithm 1 is initialized multiple times with different initial parameters for finding the clustering results that correspond to the smallest clustering error [91]. The are two ways to proceed with Steps 2 and 3. A batch version checks all N observations before prototypes are updated, and an online version updates prototypes immediately after a cluster change is encountered: one observation is moved from one cluster to another [79]. The clustering error (summed variance around centroids) is decreased for each iteration as far as convergence is reached (i.e., prototype locations do not change).

Algorithm 1. Iterative relocation algorithm

1. Select K initial centers or distributions as the initial solution.
2. Generate a new partition by using current centers.
3. Recompute new cluster centers or distributions according to the membership of clusters.
4. Repeat Steps 2 and 3 until the optimum value of the criterion function is found.

3.2 Prototype-based clustering methods

K-means is probably the most commonly used partitional clustering method. It is based on the squared Euclidean error criterion, which minimizes within-cluster sum of squared Euclidean distances. The method is also referred to as a variance minimization technique [48]. The K-means assumes continuous-valued Gaussian distributed data and creates clusters with hyperspherical shapes. It is well-known that K-means is sensitive to outliers and noise. If an observation is far away from cluster prototype, it is still connected into a cluster and, therefore, distorts the cluster shapes [91]. However, the method is simple, easy to implement, and efficient. Therefore, it is extremely popular for many kinds of cluster analysis tasks. For instance, K-means is used in the initialization of more expensive methods (e.g., minimimal learning machine [41]). The two simplest robust estimates of location are median and spatial median, whose spherical symmetric distributions are uniform and Laplace distributions, respectively. The median is the best choice as the centroid of the cluster when the type of data is discretely valued (e.g., for questionnaired data). The spatial-median is more appropriate for continuos and high dimensinal problems since its statistical efficiency improves as the number of dimensions grows [10]. The spatial-median is the multivariate generalization for the univariate median. Geometrically, the spatial median can be defined as a point of Euclidean space from which the sum of absolute distances to a given set of n points reached the minimum value. Finding the spatial median of data is a non-smooth optimization problem [50], which means that it can not be solved using classical differential calculus. However, the solution can be realized based on sequential overrelaxation (SOR) algorithm [52], which is also generalized to missing data. The spatial median is orthogonally equivalent location estimate, which makes it insensitive to all orthogonal transformations such as the rotation of a data set. The median and spatial median have the same breakdown point, which is 50%. Nevertheless, the spatial median is independent from the number of dimensions.

The general form of the clustering error function is given by

$$\mathcal{J}_p^q = \sum_{k=1}^{K} \mathcal{J}_{k,p}^q = \sum_{\mathbf{x}_i \in C_k} d(\mathbf{x}_i, \mathbf{c}_k)_p^q, \tag{10}$$

where $X = \{\mathbf{x}_i\}_{i=1}^N$ denotes data set, $d(\cdot,\cdot)$ is the distance computation or estimation strategy in the l_p^q data space, and $\{\mathbf{c}_k\}_{k=1}^K$ is the set of cluster prototypes that minimizes locally the error function and partitions the data into K disjoint subsets. $\mathcal{J}_{k,p}^q$ is the within-cluster error in cluster C_k, and l_p-norm to the q-th power is the distance measure corresponding to the different location estimates of the error function. The sample mean, median, and spatial median are obtained by choosing ($p = q = 2$), ($p = q = 1$), and ($p = 2, q = 1$), respectively. Note that if $p = 2$ or $q = 1$, the term can be omitted from the notation. Figure 2 shows gradient fields of the norms. The length of gradient vectors increases for the sample mean. Therefore, the sample mean is very sensitive to outliers and not a robust location estimate [50]. The median and spatial median are robust estimates because they depend only on the direction of data and give equal weights for all observations.

3.3 Initialization of prototype-based clustering

The prototype-based clustering methods are sensitive to the initialization, because they are based on a local-search. The methods find one locally optimal solution of an error function. The number of local optimal solutions can be large even for small data sets [80]. To avoid a poor initialization that may cause various

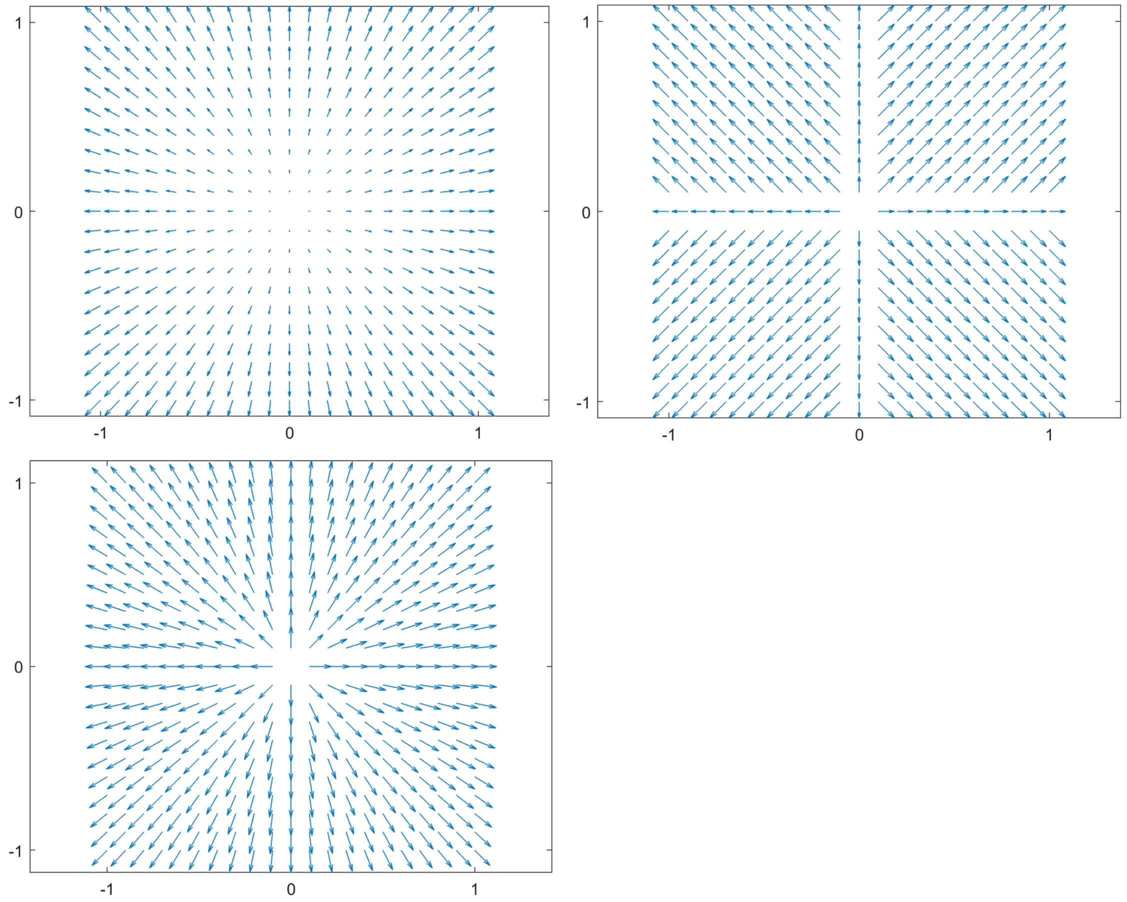


FIGURE 2 Gradient fields of $\|\mathbf{x}\|^2$ (top left), $\|\mathbf{x}\|_1$ (top right), and $\|\mathbf{x}\|$ (bottom) norms

undesired effects (e.g., sub-optimal solutions, empty clusters, and increased convergence time) the local search algorithm should be initialized carefully. Hence, the number of the repetitions of the clustering algorithm can be reduced. This is an important improvement, especially for the large-scale data sets. The repetitions are not needed at the all if there exists some deterministic heuristic to select initial points (e.g., see Article PIV).

Cluster initialization methods can be divided into three categories: random, distance optimization, and density estimation. The most used initialization in K-means is MacQueen's method [60], which selects initial prototypes at random from the data points. Therefore, the most commonly selected data points are from dense regions. This may cause initial prototypes that are selected close to each other. Similarly, Forgy's initialization method [30] is based on random selection. The method assigns data points to randomly different clusters, and the centroids of clusters are used as the initial cluster prototypes. The method lacks theoretical basis, and the clusters generated randomly may have no internal homogeneity [13]. These random methods often have poor performance [13].

The parameter-free KKZ [47] and maximin [34] initialization methods use distance optimization. The KKZ selects the vector with the maximum norm as the first prototype. Thereafter, the following prototypes are selected as the most distant to the already selected ones. The method is not computationally complex, because only one computed distance is needed for each non-prototype point at each iteration. The KKZ method is reported to be very sensitive to noise and outliers [96]. The maximin algorithm attempts to isolate the cluster prototypes that are farthest apart [34]. The algorithm randomly selects the first prototype from data points. The second prototype is the farthest point from the first prototype. The following selected prototypes are computed using minimum distances from the previously selected prototypes such as the selected point has maximum of minimum distances. The same procedure is repeated until all prototypes are selected. The algorithm works well in situations where clusters are circularly shaped and do not overlap. However, the method is sensitive to the structure of the data set and it is computationally expensive, because whenever the new prototype is selected, the distances are required to be computed for every data point from every cluster prototype.

In [7] the density initialization method for clustering was presented. The method randomly partitions data into m sub-samples and clusters each sub-sample into K clusters. The prototypes from each sub-sample are pooled into a new data set. The obtained data set is initialized m times using the prototypes of each sub-sample as initial points. The selected prototypes produce the smallest clustering error on the refined data set. In addition, the method has a robust variant, which is insensitive to erroneous data values [96].

Currently, the most popular algorithm for initialization is K-means++ [2]. It uses probability distribution based on distances to already selected nearest centroids. The computational complexity of K-means++ is equal to the complexity of K-means, which is linear. The selection time of initial centroids is slightly higher compared to K-means. However, on average, the overall algorithm is faster because it usually converges with fever iterations. In addition, the K-means++ is proven to be more accurate than the K-means regarding clustering errors [2]. The method is given in Algorithm 2. Note that the selected distance metric is required to be defined beforehand.

Algorithm 2. K-means++ type of initialization

1. Choose center $\mathbf{c}_1$ uniformly random from $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N,\ \mathbf{x}_i \in \mathbb{R}^n$.
2. Choose the next center $\mathbf{c}_j = \mathbf{x}_i \in \mathbf{X}$ with probability

$$\frac{\min\limits_{k=1,\dots,j-1} d(\mathbf{x}_i, \mathbf{c}_k)_p^q}{\sum\limits_{x \in C_k} \min\limits_{k=1,\dots,j-1} d(\mathbf{x}, \mathbf{c}_k)_p^q}.$$

3. Repeat Step 2 until K centers is chosen.

3.4 Distance estimation in clustering

Let us follow the same assumptions as given in Section 2.2.4. Hence, data values are from the conditional multivariate normal distribution where the missing values are conditioned with the observed ones (i.e., missing values are MAR). Two-stage clustering algorithm based on the estimated and computed distances is given in Algorithm 3.

Algorithm 3. Clustering based on distance estimation

1. Select initial prototypes using K-means++ algorithm (see Algorithm 2).
2. Compute the statistical parameters ($\boldsymbol{\mu}$ and Σ) of the conditional multivariate Gaussian distribution using the EM method with the maximum negative log-likelihood convergence criterion.
3. Estimate distances using Equation (5) or (7) while performing the iterative refinement phase of the clustering (Steps 2–4 in Algorithm 1).
4. Repeat the iterative refinement phase using the available data strategy in the distance computation (see Equation (3)).

Note that Algorithm 3 convergences twice: the first time with the distance estimation and the second time without the estimation. Based on the experiments in PIII, the clustering with distance estimation produced prototypes closer to the real centroids than the clustering without estimation. However, even more accurate results were obtained by giving these estimation-based prototypes as initial values to the clustering based on the available data strategy (Step 4 in Algorithm 3).

4 CLUSTER VALIDATION

Cluster validation, or clustering evaluation, is a challenging but important task in cluster analysis. Finding the optimal number of clusters (K) given as an input to the clustering algorithm, is essential because the number is rarely obvious. Almost every clustering algorithm will find clusters in a data set even if the data have no clustering tendency. In this situation, a validation measure is needed to describe how good clustering is and what may be the optimal number of clusters.

The quality of the clustering results is commonly validated with cluster validation indices (CVIs). In general, cluster validation indices have three different criteria: internal, relative, and external [43]. Internal indices evaluate the quality of the clustering result using the data alone. Relative indices compare multiple clustering structures (generated by different parameters, for example) and decide which of them is better in some sense. In [36], relative indices were categorized to internal indices, because there is only a slight difference in the definitions of these indices. External indices measure the performance by comparing the obtained clustering structure to the correct structure (ground true) if the real cluster labeling is available. Indices can be used for measuring cluster stability as an amount of variation in the clustering solution over different sub-samples drawn from input data.

The cluster validation indices plot the number of clusters against measured index values. There are multiple ways to detect the optimal number of clusters. The most straightforward way is to use the global minimum or maximum of the index curve. However, some cluster validation indices (e.g., Calinki-Harabasz [11] and Ray-Turi [75]) recommend to use the first local minimum or maximum. The knee point is determined as the turning point of the curve (from optimal to suboptimal direction). In addition, there are cluster validation measures, which are only monotonily increasing or decreasing. In these cases, the most significant local change could be observed on the curve, which is called knee or jump point. There exist multiple ways to determine the knee point. The classical way is to indicate the change in measured index values with every increase in the number of clusters [85]. Ideally the change can be validated through significance testing, which requires some assumptions like the normality of the distributions [85]. The are also more sophisticated heuristics to localized knee points. For example, in [95], a reliable method to identify the knee points based on Bayesian information criterion was proposed. The experiments supported the performance of the offered approach over the performance of many conventional approaches. In [78], a knee point method was proposed, which defined the curvature of continuous and discrete data sets at any point as a function of the first and second derivatives. The method is applicable to a wide range of systems, including online and offline types of applications.

In determining the optimal number of clusters, the parameter K is optimized and other parameters are fixed. This contains the definition of the fixed range of clusters $[K_{min}, K_{max}]$ and the basic procedure involves the steps given in Algorithm 4. Note that there is no one best index that works with all kinds of clustering methods and data sets. Hence, comparing the results of multiple indices is highly recommended. For example, depending on the clustering algorithm, the internal indices are applied to hard (crisp) or soft (fuzzy) clustering. In addition, the data set may include noise, different densities, cluster overlap, subclusters, and skewed distributions, etc. [59]. For this purpose, many comparisons of indices have been made [1, 56, 65].

Algorithm 4. Determination of the optimal number of clusters

1. Select a data set $\mathbf{X}$ and repeat a clustering algorithm successively, ranging the number of clusters from K_{min} to K_{max}.
2. Obtain prototype-based clustering results (labels of each observation and cluster centroids) and compute internal index values for each.
3. Identify the best result using global minimum or maximum, or use knee points to localize local minimums and maximums.
4. Test the validity goodness of the solution by using external indices if ground true of the partition is known.

4.1 Internal quality measures

The optimal number of clusters can be determined using internal CVIs with different K values as an input parameter to the clustering algorithm. The computation of internal CVIs are commonly based on the compactness and separation of the clusters. In a good clustering solution, the within-cluster similarity (Intra) is low and the between-cluster separability (Inter) is high. Usually, the division between Intra and Inter is applied and the measured value is at minimum or maximum based on the order of the division.

Let us define the basic notations followed in the rest of this section. The centroid of the whole data (mean, median, or spatial median) is $\mathbf{m}$. The total clustering error, the within-cluster error, and the number of observations in the cluster $\mathbf{C}_k$ are denoted by $\mathcal{J}$, $\mathcal{J}_k$, and n_k, respectively (see Eq. (10)). Note that it is assumed that $\mathcal{J} = \mathcal{J}_2^1$ and $\mathcal{J}_k = \mathcal{J}_{k,2}^1$ (i.e., the clustering error is the sum of the Euclidean distances if otherwise is not specified). Since observations may consist of missing values, distance computation or estimation strategy is always needed for computing Intra and some indices for computing Inter (denoted by $d(\cdot,\cdot)$). Depending on the formula of CVI, the measure attempts to be minimized (denoted by $\Downarrow$) or maximized (denoted by $\Uparrow$).

Calinki–Harabasz: Calinski-Harabasz (CH) index is also known as variance ratio criterion, since the variance between prototypes is aimed to be maximized, and between observations and their local prototypes are attempted to be minimized [11]. The method is originally based on the squared Euclidean distance. The index is defined as:

$$\mathrm{CH}^{\Uparrow} = \frac{\sum\limits_{k=1}^{K} n_k \|\mathbf{c}_k - \mathbf{m}\|^2}{(K-1)} \Big/ \frac{\mathcal{J}^2}{(N-K)}.$$

Davies–Bouldin: The Davies-Bouldin (DB) index is defined by the average of cluster evaluation measures for all the clusters. In addition, the index attempts to use as low Inter cluster separation as possible [18]. The index is defined as:

$$\mathrm{DB}^{\Downarrow} = \frac{1}{K} \sum_{k=1}^{K} \max_{\substack{k' \neq k \\ k^* \neq k}} \left(\left(\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}} \right) / \|\mathbf{c}_k - \mathbf{c}_{k^*}\| \right), \quad k', k^* = 1, \ldots, K.$$

Davies–Bouldin*: The Davies-Bouldin* (DB*) is an extended version of the original DB algorithm, which applies maximization and minimization separately [49]. Hence, the minimum Inter cluster separation is guaranteed in the solution. The DB* is defined as:

$$\mathrm{DB}^{*\Downarrow} = \frac{1}{K} \sum_{k=1}^{K} \max_{k' \neq k} \left(\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}} \right) / \min_{k^* \neq k} \|\mathbf{c}_k - \mathbf{c}_{k^*}\|, \quad k', k^* = 1, \ldots, K.$$

Generalized Dunn: The Generalized Dunn (GD) indices are improvements of the Dunn index, which are less sensitive to noisy observations [3]. They include three definitions for within-cluster distances and six definitions of the between cluster distances, which leads to 18 definitions. The following definition uses the maximum average within-cluster diameter as Intra. The Inter is defined as the minimum distance between cluster centroids:

$$\mathrm{GD}^{\Uparrow} = \min_{k' \neq k} \|\mathbf{c}_k - \mathbf{c}_{k'}\| / \max \left(2 \times \frac{\mathcal{J}_k}{n_k} \right), \quad k, k' = 1, \ldots, K.$$

Experimental results suggest that Inter cluster separation has a more important effect in cluster validation than cluster diameter [3].

kCE-index: The kCE-index (KCE) uses only the total sum of within-cluster errors for determining the optimal number of clusters, and therefore, index is able to recognize the case of one cluster or no clusters where other indices usually suggest large numbers [45]. This is advantageous over other internal CVIs. The number of cluster, K, is used as a weight of the squared clustering error. The KCE is defined as follows:

$$\text{KCE}^{\Downarrow} = K \times \mathcal{J}^2.$$

Pakhira–Bandyopadhyay–Maulik: The Pakhira-Bandyopadhyay-Maulik (PBM) index is a product of three factors [71]. The first factor indicates divisibility of a K cluster system. The factor reduces with an increase in K. The second factor is a measure of the compactness of a K cluster system, which is attempted to increase. The third factor is maximum Inter cluster separation and this is attempted to increase. So while the first factor is decreasing, the other two are increasing with an increase in K. The PBM is defined as follows:

$$\text{PBM}^{\Uparrow} = \left(\frac{1}{K} \times \left(\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{m}\| / \mathcal{J} \right) \times \max_{k' \neq k} \|\mathbf{c}_k - \mathbf{c}_{k'}\| \right)^2, \quad k, k' = 1, \dots, K.$$

The PBM index works for both crisp and fuzzy clustering. The index ensures the formation of a small number of compact clusters with a large separation between at least two clusters.

Ray–Turi: The Ray-Turi (RT) index comes from image segmentation [75]. The index was initially evaluated with synthetic images for which the number of clusters was originally known. It was also implemented for natural images. The index was developed for K-means clustering and, therefore, follows the notation of the squared norm. The RT takes the average value of distances from observations to their local centroids and use the minimum distance between cluster centroids. The definition is the following:

$$\text{RT}^{\Downarrow} = \frac{1}{N} \times \frac{\mathcal{J}^2}{\min\limits_{k' \neq k} \|\mathbf{c}_k - \mathbf{c}_{k'}\|^2}, \quad k, k' = 1, \dots, K.$$

Silhouette: The silhouette (SIL) measure for each observation describes how similar an observation is to other observations in the same cluster, compared to observations in other clusters [48]. The silhouette value is defined as:

$$\text{SIL}^{\Uparrow} = \frac{1}{N} \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in \mathbf{C}_k} \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{\max(a(\mathbf{x}_i), b(\mathbf{x}_i))},$$

where $a(\mathbf{x}_i)$ is the average distance from ith observation to other observations in the same cluster:

$$a(\mathbf{x}_i) = \frac{1}{n_k - 1} \sum_{\mathbf{x}_j \in \mathbf{C}_k} d(\mathbf{x}_i, \mathbf{x}_j)_2.$$

and $b(\mathbf{x}_i)$ is the minimum average distance between ith observation and observations in a different cluster minimized over the clusters:

$$b(\mathbf{x}_i) = \min_{k' \neq k} \frac{1}{n_{k'}} \sum_{\mathbf{x}_j \in \mathbf{C}_{k'}} d(\mathbf{x}_i, \mathbf{x}_j)_2, \quad k' = 1, \ldots, K.$$

Note the $d(\cdot,\cdot)$ notation in equations. Distances are required to be computed through distance computation (see Section 2.2.3) or distance estimation strategy (see Section 2.2.4) because data vectors may consist of missing values.

The average silhouette value is in the range of $[-1, 1]$. A high index value indicates that observations are well matched to their own clusters and not matched to other clusters. A negative index value indicates that the clustering solution has too many or too few clusters. If the silhouette value is equal to zero, the data set may have overlapping clusters.

WB-index: The WB-index (WB) can be categorized with the sum-of-squares-based indices and it has a similar trend as the inverse of the CH index [94]. However, the CH index may affect the data size N, because when N is very high the weight factor $\frac{M-1}{N-M}$ dominates over the quotient of the Inter cluster separation and within cluster compactness. In addition, the CH index may not detect accurately the highly overlapping clusters as separate clusters. Hence, the WB could be a better choice in some specific data sets. The index is defined as:

$$\mathrm{WB}^{\Downarrow} = \frac{K \times \mathcal{J}^2}{\sum\limits_{k=1}^{K} n_k \|\mathbf{c}_k - \mathbf{m}\|^2}.$$

Wemmert–Gançarski: The Wemmert-Gançarski (WG) forms measure for each observation, which is the quotient between the distance of this observation to the centroid the observation belongs to and the smallest distance of this observation to the centroid of all the other clusters [21]:

$$\mathrm{WG}^{\Downarrow} = \frac{1}{N} \sum_{k=1}^{K} \max \left\{ 0, n_k - \sum_{\mathbf{x}_i \in C_k} \frac{d(\mathbf{x}_i, \mathbf{c}_k)_2}{\min\limits_{k' \neq k} d(\mathbf{x}_i, \mathbf{c}_{k'})_2} \right\}, \quad k' = 1, \ldots, K.$$

Regarding the formula, if the quotient is greater than n_k, it is ignored. Intra and Inter require the use of distance computation or estimation strategy in case of missing values in data.

Internal indices in general fashion: Table 1 shows internal cluster validation indices in general forms based on l_p^q-norm settings. All of the indices are aimed to be minimized, because the ordering of division is $\frac{\text{Intra}}{\text{Inter}}$. In addition, the simplified presentations of CVIs are given. Additional constant terms and extra functions which do not affect to the functionality of indices are removed from the final presentation.

TABLE 1 Internal cluster validation indices in general fashion

Index	Intra	Inter	Formula
CH	$\mathcal{J}_p^p$	$\sum\limits_{k=1}^{K} n_k \lVert \mathbf{c}_k - \mathbf{m} \rVert_p^p$	$\frac{K-1}{N-K} \times \frac{\text{Intra}}{\text{Inter}}$
DB	$\frac{\mathcal{J}_{k,p}^q}{n_k} + \frac{\mathcal{J}_{k',p}^q}{n_{k'}}$	$\lVert \mathbf{c}_k - \mathbf{c}_{k^*} \rVert_p^q$	$\frac{1}{K} \sum\limits_{k=1}^{K} \max\limits_{k \neq k'} \frac{\text{Intra}(k,k')}{\text{Inter}(k,k')}$
DB*	$\frac{\mathcal{J}_{k,p}^q}{n_k} + \frac{\mathcal{J}_{k',p}^q}{n_{k'}}$	$\lVert \mathbf{c}_k - \mathbf{c}_{k^*} \rVert_p^q$	$\frac{1}{K} \sum\limits_{k=1}^{K} \frac{\max\limits_{k \neq k'} \text{Intra}(k,k')}{\min\limits_{k \neq k^*} \text{Inter}(k,k^*)}$
GD	$\max \frac{\mathcal{J}_{k,p}^q}{n_k}$	$\min\limits_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p^q$	$\frac{\text{Intra}}{\text{Inter}}$
KCE	$\mathcal{J}_p^p$	1	$K \times \text{Intra}$
PBM	$\mathcal{J}_p^q$	$\max\limits_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p^q$	$\frac{K \times \text{Intra}}{\text{Inter}}$
RT	$\mathcal{J}_p^q$	$\min\limits_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p^q$	$\frac{\text{Intra}}{\text{Inter}}$
SIL	$\frac{1}{n_k - 1} \sum\limits_{\mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j)_p^q$	$\min\limits_{k \neq k'} \frac{1}{n_{k'}} \sum\limits_{\mathbf{x}_j \in C_{k'}} d(\mathbf{x}_i, \mathbf{x}_j)_p^q$	$\sum\limits_{k=1}^{K} \sum\limits_{i=1}^{N} \frac{\text{Inter}(\mathbf{x}_i) - \text{Intra}(\mathbf{x}_i)}{\max(\text{Intra}(\mathbf{x}_i), \text{Inter}(\mathbf{x}_i))}$
WB	$\mathcal{J}_p^p$	$\sum\limits_{k=1}^{K} n_k \lVert \mathbf{c}_k - \mathbf{m} \rVert_p^p$	$K \times \frac{\text{Intra}}{\text{Inter}}$
WG	$d(\mathbf{x}_i, \mathbf{c}_k)_p^q$	$\min\limits_{k \neq k'} d(\mathbf{x}_i, \mathbf{c}_{k'})_p^q$	$\sum\limits_{k=1}^{K} \sum\limits_{\mathbf{x}_i \in C_k} \frac{\text{Intra}(\mathbf{x}_i)}{\text{Inter}(\mathbf{x}_i)}$

4.2 External quality measures

External indices include different measures of variation, which can be used to obtain different stability indicators.

Accuracy: Accuracy (ACC) is the simplest stability measure. The method computes quotient of the total number of correctly predicted cluster labels and the total number of cluster labels [86].

Adjusted Rand index: Adusted Rand index (ARI) is a pair-counting-based measure. The index counts pairs of labels on which two clusterings agree or disagree [88]. Note that the minimum value of the index can yield negative values if the index is less than the expected index. The maximum value is one, which indicates the best solution of the clustering.

Given a set $\mathbf{X}$ of N observations, and two partitions of these elements, namely $U = \{U_1, U_2, \ldots, U_r\}$ and $V = \{V_1, V_2, \ldots, V_s\}$, the overlap between U and V can be summarized as in Table 2, where each entry n_{ij} denotes the number of observations in common between U_i and V_j, $n_{ij} = |U_i \cap V_j|$, $U_i \cap U_j = \emptyset$, $V_i \cap V_j = \emptyset$ for $i \neq j$. The ARI using a permutations model is defined as:

$$\text{ARI} = \frac{\Sigma_{ij}\binom{n_{ij}}{2} - [\Sigma_i\binom{a_i}{2}\Sigma_j\binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\Sigma_i\binom{a_i}{2} + \Sigma_j\binom{b_j}{2}] - [\Sigma_i\binom{a_i}{2}\Sigma_j\binom{b_j}{2}]/\binom{n}{2}},$$

TABLE 2 Contingency table

U/V	V_1	V_2	$\dots$	V_s	Sums
U_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
U_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Sums	b_1	b_2	$\dots$	b_s	$\sum_{ij} n_{ij} = N$

where n_{ij}, a_i, and b_j are values from the contingency table.

Normalized mutual information: Mutual information computes amount of information obtained from random variable by observing the other random variable. In telecommunications, the capacity of the channel is measured by using mutual information [84]. Normalized mutual information (NMI) is an information theoretic measure with normalization property, i.e., the measure lies within a fixed range $[0, 1]$ [88]. Mutual information is a symmetric measure and it determines the statistical information shared between two distributions. It characterizes reduction in the entropy which is obtained knowing the actual cluster labels. There are many variants of the index that use different normalization terms for the mutual information. Joint entropy, minimum entropy, maximum entropy, average entropy, and squared entropy are commonly used divisors. Entropy is a fundamental notion in an information theory that computes the expected amount of information held in a random variable. The mutual information normalized by the average entropy is defined as:

$$\text{NMI} = \frac{I(U,V)}{(H(U) + H(V))/2},$$

where $I(\cdot,\cdot)$ is mutual information of cluster labels and $H(\cdot)$ denotes entropy of labeling. Many external CVIs assume identical number of clusters between clustering solutions. However, NMI overcomes this assumption using the normalization term. Hence, NMI is a promising external measure for determining the quality of the clustering.

5 SUMMARY OF THE INCLUDED ARTICLES

5.1 PI: Game learning analytics for understanding reading skills in transparent writing system

Article PI was published in the *British Journal of Educational Technology*, 51(6):2-376–2390, 2020

Background: Digital serious games provide an alternative to traditional teaching methods like classroom lessons. Serious games analytics are designed for measuring, collecting, analyzing, and reporting data about serious games learners. The main objective is to improve learning and tailor learning environment for various skilled learners. In the study, serious games analytics were applied to the Finnish version of *GraphoLearn*. The game was designed to support the decoding skills of Finnish children with difficulties in reading by helping to connect speech sounds to their written counterparts (i.e., letters). The study offered a profiling tool that used missing values handling, clustering, and cluster validation for identifying the distinct profiles based on players' game log data, informing which choices learners have made in the game.

Methods and data: The data was gathered from 1,632 players who were 6.5–8.75 years old. In total, 1,275 learners played with lowercase letters and 357 learners used uppercase letters. The learners' actions during the game were logged into a database. The most interesting information to be logged were learners' inputs, time spent with each task, playing times, and interval times between playing sessions. The game presented 23 target letters and speech sounds each of which was introduced only one time. The collected data included 4.66% missing values, because of unanswered tasks.

The players were divided into two groups based on whether they used lowercase or uppercase letters. All combinations related to correct and incorrect responses obtained from all learners produced binary feature vectors, each consisting of 529 elements. However, the computation cost was reduced by filtering out the features which did not include a noticeable number of erroneous selections. Clustering and cluster validation for lowercase and uppercase data sets were performed using the PDS method in the distance computation. The cluster validation was performed, applying the classical knee-point method to the curves of indices. The speed of improvement of index values was the main interest. The knee-points of the validation curves were statistically analyzed with Wilcoxon's rank-sum test.

Results and conclusions: The most common errors were related to confusing phonetically and visually similar letters. For instance, the letter n was often mixed to letters h and m, and the sound */g/* was mixed with the sounds */d/* and */k/*. The results revealed six profiles for lowercase letters, one "high" performing, three "medium" performing, and two "low" performing profiles, and five profiles for uppercase letters. The measured meta parameters in different profiles were error rates, progressions, total playing times, and interval times. The learners in the weakest performing profile mixed the most of the target letters. However, the players in the two weakest profiles showed the best progression while playing the game, which suggests that the combination of *GraphoLearn* and school-provided reading instructions help children who have difficulties for mastering letter-sound pairs. The learners achieved the better results with uppercase letters, which may be because uppercase letters are visually less similar. In general, the study offered the profiling tool for identifying different types of learners in an alphabetical learning game. The tool was developed for letters but it can be extended for larger units such as syllables or words. Further research is required for repeating the clustering at regular time intervals to see players division into varied skilled profiles and to monitor learner's development in the game by detecting their connections to the profiles.

5.2 PII: Comparison of cluster validation indices with missing data

Article PII was published in the *proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning*, pages 461–466, 2018.

Background: The aim of cluster analysis is to evaluate the data structures based on the multiple clustering solutions. Different cluster validation indices have been developed and compared for finding the optimal number of clusters. However, missing values are rarely considered in the evaluation process.

Methods and data: The study presented a prototype-based K-means and K-medians clustering algorithms and provided the comparison of ten well-performing internal CVIs. The clustering methods and all the indices were developed to be tolerant for missing values in data. The computation of the clustering and cluster validation is mainly based on the computation of the distances between data vectors. The results may not be accurate if distances are computed between incomplete data. Therefore, the PDS for computing scaled pairwise distances was adopted to the current study. The Euclidean distance based PDS and all the CVIs, originally based on squared or non-squared Euclidean distances, were extended to the City block distance. Eight synthetic data sets were selected from previous studies and two new data sets were created for the current study. All data sets were scaled at the range of $[-1, 1]$. The predefined percentages of missing values were generated and data were MCAR. The initialization of clustering was performed in an iterative manner, benefiting the previously selected centroids and using the generalized version of K-means++ algorithm (see Algorithm 2).

Results and conclusions: Concerning the results, WG, SIL, and KCE were generally the best performing indices recommending 64/80, 63/80, and 58/80 total correct solutions, respectively. All data sets were continuous valued, which may explain the better results of the indices with the Euclidean distance that is mainly based on the assumption of the Gaussian distributed data. The best performing indices with the Euclidean distance were WG, KCE, and CH, suggesting 36/40, 33/40, and 33/40 correct solutions, respectively. Regarding the different degrees of missing values, the stability of the indices was measured. The most stable indices for the Euclidean distance were WG and DB*, whereas CH was the most stable for the City block distance. These indices always recommended the identical number of clusters with the Euclidean or City block distance even though the number of missing values varied. In general, most of the indices decreased the performance when the degrees of missing values were increased in the data sets.

5.3 PIII: Improving clustering and cluster validation with missing data using distance estimation methods

Article PIII was published in the *Computational Sciences and Artificial Intelligence in Industry: New Digital Technologies for Solving Future Societal and Economical Challenges*, pages 123–133, 2022.

Background: Missing values in data is a common problem in the real world and it is rarely considered in clustering and cluster validation tasks. Very often, observations with missing values are omitted. The *k*NNI imputation and the distance computation methods (ADS, PDS) are alternative treatments for missing values in many machine learning tasks. Recently two promising distance estimation methods, ESD and EED, were presented and utilized for incomplete data. The methods assume that the missing values are random variables from a conditional multivariate normal distribution where missing values are conditioned with observed ones. The distance estimation methods have provided more accurate results compared to traditional methods in the experiments based on computation of distances between observations within data sets consisting of a wide range of missing values. However, the methods were not investigated in practical machine learning problems. The current study applied the EED method for non-squared clustering and the ESD or the EED was utilized to the indices depending on the original squared or non-squared formulas of indices.

Methods and data: In the study, the robust K-spatialmedians clustering was utilized. The clustering algorithm minimized the sum of the Euclidean distance error function and was based on ADS distance computations. The ADS approach omits missing values using binary projection matrix, which presents sparsity patterns of each observations. The clustering method was compared to the traditional K-means clustering and distance estimated K-spatialmedians clustering. The reference results of K-means clustering were obtained from the previous study.

Results and conclusions: The results show that the root mean square errors (RMSE) to the real centroids were lower with the K-spatialmedians and EED-based distance estimation compared to the ADS-based estimation over 100 repetitions of the replicated clustering (with 100 replicates) and generation of the missing values. In addition, the results were even more accurate when the obtained distance-estimated prototypes were used as starting points for the K-spatialmedians clustering with the ADS. Regarding the computational complexity, the EED-based clustering showed almost equal computation times compared to the ADS-based clustering with larger data sets, and it appeared to be faster with smaller data sets.

Most of the CVIs improved the performance only by changing the clustering method from K-means to more robust K-spatialmedians. Many indices did not increase the performance based on the distance estimation and these results were not reported. However, in general, the indices obtained the better performance by using the novel two-stage clustering method based on the estimated distances and the pipelined results to ADS-based clustering. Most especially, better results were obtained with the maximum degrees of missing values in the synthetic data sets.

5.4 PIV: Toolbox for distance estimation and cluster validation on data with missing values

Article PIV was published in the *IEEE Access*, 10:352–367, 2021.

Background: The study provided the toolbox for data preprocessing, distance estimation, clustering, and cluster validation in the presence of missing values in data sets. The methodological background of implemented methods was introduced. In addition, the study proposed descriptions of the core functions and offered the use case examples of the basic algorithms in the toolbox.

Methods and data: The experiments were divided into three parts. In the first part, the missing values were generated to seven real-world data sets, selected from the UCI repository, and the accuracies of the implemented missing values computation or estimation methods were measured using the RMSE errors to the real distances between observations in data sets. The mean values and standard deviation were recorded in 250 repetitions. The implemented distance algorithms were validated against the reference methods given in the previous studies. The results suggest that the ESD distance estimation is better than the EED estimation, because the relative differences in the accuracies between these methods were small and the ESD method is computationally less complex.

The second part compared K-spatialmedians clustering algorithms with and without distance estimation using ten internal CVIs. The quality of clustering results was validated using three external CVIs. In addition, the key point selection method was utilized in the initialization of the clustering. The cluster validation results were computed over eight synthetic and three real-world data sets, which consisted of completely randomly generated missing values. The experiments related to the initialization of clustering with the key point selection algorithm assumed data sets were complete and in 2D. Hence, ICkNN (k =2) imputation was applied to the incomplete data sets, and the real-world data were scaled to 2D using multidimensional scaling (MDS) algorithm. The reference results were obtained from a previously published study. The achieved results with the key point selection were competitive to the reference results.

In the third part, the performance of the indices was computed for multidimensional data. For this purpose, a total of 12 multidimensional data sets, which consisted of 15 cluster centroids, various dimensions, different percentages of missing values, and various degrees of cluster overlap were generated using the data set generator, which was recently presented in the previous publication. Based on the results, many of the indices were able to find the correct number of clusters for the most of data sets in their original dimensions. The 2D projection of data sets was also experimented. However, the overall results of the indices were significantly worse in 2D. In the original multidimensional presentation, three of the indices did not find any correct choices, including the KCE that uses only Intra term for the computation of the index values. These findings suggest that the Inter term separates clusters better in multidimensional space, and the clustering errors between good and bad clustering results in a high-dimensional space is small. The WG index performed the best with the multidimensional data.

Results and conclusion: Even though the paper presented a large amount of experiments, the main objective of the study was to provide an easy-to-use toolbox for researchers and practitioners to build various pipelines from missing values handling and data preprocessing to cluster analysis and model validation. The versatile functionality of the toolbox allows its usability for other machine learning tasks as well like supervised learning. Future work is required for identifying a correct heuristic to the key point selection algorithm, because clustering algorithms are sensitive for accurate initialization.

6 CONCLUSIONS AND FUTURE WORK

This dissertation was composed of four peer-reviewed articles, which are focused on prototype-based clustering and internal cluster validation on data with missing values. Contributions from each article are summarized in Table 3. These data mining methods perform a central part in the KDD process. The missing values are common in real-world applications and are often not considered in the development work of new data mining algorithms. The worst-case scenario is that the implemented methods do not work if the training data contains missing values. There were many reasons for selecting prototype-based clustering in grouping the objects: the methods are usually well-defined, straightforward to implement, computationally less complex, and produce interpretable geometrically closed subsets as a clustering result. In addition, in cluster validation, internal validation indices follow a similar notation of clusters as prototype-based methods with the appropriately selected clustering error criterion (i.e., clusters need to be compact and separable).

Article PI provides a profiling tool based on clustering and cluster validation. The target application is *GraphoLearn* serious game, which is developed to build connections between speech sounds and their written targets. The study focused on the smallest imaginable units in the Finnish language, namely phonemes and their written equivalents, letters. The learners were first-grade students with difficulties in learning to read. The learners' responses during the game were logged into a database. The data included missing values because of unanswered tasks. The profiling tool identified six varied skilled profiles. The players in the weakest profile mixed almost all target sounds with incorrect letters. However, these players showed the best progression while playing the game, which suggests that the combination of *GraphoLearn* and school-provided reading instruction helps children who have difficulties in reading. Articles PII and PIII present distance estimation methods for treating missing values in clustering and cluster validation indices. The best results were obtained by using two stages K-spatialmedians clustering algorithm. The first phase estimated the distances between observations and cluster prototypes using the EED distance estimation method. In the second phase, the obtained prototypes were given as an in-

TABLE 3 Summary of contributions

Article	Contribution
PI	1) Provided a profiling tool for serious games learners 2) Utilized internal CVIs in selecting the number of learners' profiles in a *GraphoLearn* 3) Detected the knee-points of the internal CVI curves using visualizations of the CVI results and the Wilcoxon's rank-sum test 4) Analyzed the obtained learners' profiles 5) Proposed visualized presentations of confused letters in distinct profiles
PII	1) Presented the PDS method and its extension to City block distance 2) Implemented ten well-performing internal CVIs and extended them to the missing values and City block distance 3) Compared the City block and squared Euclidean distance based cluster validation results using the synthetic data sets with predefined percentages of missing values
PIII	1) Proposed a brief introduction to the ADS, ESD, and EED distance computation or estimation strategies 2) Utilized the strategies to K-spatialmedians clustering and internal CVIs 3) Presented a novel two-stage distance-estimation based clustering method and evaluated its performance against reference methods
PIV	1) Implemented a toolbox for missing values handling, data preprocessing, clustering, and cluster validation 2) Provided the theoretical background of the implemented methods 3) Proposed use-case examples of the basic use of the toolbox 4) Validated the implemented methods against the reference methods in the previous studies 5) Evaluated the clustering solutions using the external validation measures 6) Presented new experiments related to the initialization of the clustering and to the cluster validation

put for initializing K-spatialmedians clustering with the ADS distance computation method. The computational complexity of two stages clustering was almost equal to the complexity of traditional K-spatialmedians clustering without distance estimation when 2D synthetic data sets were used in the experiments. Article PIV proposes a toolbox that consists of the functionality for handling missing values in the data preprocessing, clustering, and cluster validation. The descriptions of all methods and use case examples are provided. An easy-to-use toolbox is freely available for researchers and practitioners online[1].

A limited number of data sets were used in the studies. Hence, future work should develop new, more complex synthetic data sets with different dimensionalities similar to [42] and [32]. In addition, the developed methods could be evaluated with a broader range of real-world data. Since the data volumes are increasing every day, more scalable methods are required for analyzing big data. Parallel computation and efficient use of memory capacity are important aspects that should be noticed in future research considering missing values handling and cluster analysis. Article PIV introduced a new initialization method based on selecting key points associated as initial points to the clustering. The selected key points included observations with relatively higher densities and density-based distances than others (see Article PIV for more details). The key point method introduced good solutions for most data sets with varying portions of missing values. However, in some cases, the method could not to recognize lower density clusters or key points located near the same cluster. Hence, future improvements for the key-point-based initialization algorithm are required to support a higher diversity of data.

The provided distance estimation strategies were capable of solving clustering problems. In addition, many other machine learning methods are based on computed distances. Hence, the developed methods are applicable, for example, supervised learning [14, 41, 17, 9].

[1] https://github.com/markoniem/nanclustering_toolbox

YHTEENVETO (SUMMARY IN FINNISH)

Puutteellisen datan klusteroinnin validointi

Tämä neljän vertaisarvioidun artikkelin väitöskirja keskittyi prototyyppipohjaiseen klusterointiin ja klusteroinnin sisäiseen validointiin datajoukoilla, jotka sisälsivät puuttuvia arvoja. Nämä tiedonlouhinnan menetelmät ovat keskeisessä osassa tietämyksen muodostamisprosessissa. Reaalimaailman sovelluksissa puuttuvat arvot ovat yleisiä ja usein ne jätetään huomioimatta algoritmikehityksessä. Pahimmassa tapauksessa toteutetut algoritmit eivät toimi ollenkaan, mikäli data sisältää puuttuvia arvoja. Prototyyppipohjaiset klusterointimenetelmät ovat usein hyvin määriteltyjä, suoraviivaisia toteuttaa, laskennallisesti tehokkaita ja tuottavat havainnollisia, geometrisesti yhdenmukaisia osajoukkoja klusterointitulokseksi. Tämän lisäksi klusteroinnin validoinnissa sisäiset indeksit noudattavat samankaltaista notaatiota klustereista. Klusterien täytyy olla selkeästi määriteltyjä, tiiviitä ja toisistaan erottuvia.

Ensimmäisessä artikkelissa esitetään profilointisovellus, joka pohjautuu klusterointiin ja klusteroinnin validointiin. Sovelluskohteena on *GraphoLearn*-niminen lukemaan opettelu -peli ja kohdejoukkona koulunsa vasta aloittaneet lapset, joilla on ollut vaikeuksia lukemaan opettelussa. Lapsien vastaukset kirjain-äännetehtäviin tallennettiin tietokantaan ja kerätty data sisälsi myös puuttuvia arvoja vastaamattomista tehtävistä. Profilointityökalulla pystyttiin tunnistamaan yhteensä kuusi eritasoista profiilia. Heikoimman profiilin lapsilla oli vaikeuksia tunnistaa lähes kaikkia kirjaimia äänteiden perusteella. Myöhäisemmässä vaiheessa profiilissa tapahtui kuitenkin eniten kehitystä, joka tukee pelin ja kouluopetuksen vaikuttavuutta lukemaan oppimisessa. Artikkelit kaksi ja kolme esittävät etäisyyksien estimointeihin perustuvia puuttuvan datan käsittelymenetelmiä klusteroinnissa ja klusteri-indekseissä. Parhaat tulokset saavutettiin käyttämällä kaksiosaista K-spatialmedians-klusterointia. Ensimmäisessä vaiheessa etäisyydet havaintojen ja klusteriprototyyppien välillä estimointiin EED-menetelmällä. Toisessa vaiheessa saatuja prototyyppejä käytettiin klusteroinnin alustuksessa perinteisessä ADS-etäisyyslaskentaan perustuvassa K-spatialmedians-klusteroinnissa. Kaksiosaisessa klusteroinnissa ei havaittu ylimääräistä laskennallista raskautta kokeissa käytetyillä synteettisillä 2D-datoilla. Viimeisessä artikkelissa kaikki kehitetyt menetelmät on koottu yhdeksi ohjelmistokokonaisuudeksi, jolla voi käsitellä puuttuvia arvoja datan esiprosessoinnissa, klusteroinnissa ja klusterivalidoinnissa. Artikkeli sisältää kaikkien menetelmien kuvaukset ja tarjoaa lisäksi käyttöesimerkkejä. Ohjelmisto on avoimesti saatavilla[2] ja se on suunniteltu erityisesti tutkimuskäyttöön, mutta myös aiheesta kiinnostuneille harrastelijoille.

Esitetyt puuttuvan datan etäisyysestimointimenetelmät osoittivat toimivuutensa klusteroinnissa. Tämän lisäksi on olemassa suuri määrä muita koneoppimismenetelmiä, jotka myös perustuvat laskettuihin etäisyyksiin. Kehityt menetelmät soveltuvat esimerkiksi ohjattuun oppimiseen [14, 41, 17, 9].

[2] https://github.com/markoniem/nanclustering_toolbox

REFERENCES

[1] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern recognition*, vol. 46, no. 1, pp. 243–256, 2013.

[2] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[3] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 301–315, 1998.

[4] J. A. Bilmes, "A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[5] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, Verlag, 2006.

[6] P. Blikstein, M. Worsley, C. Piech, M. Sahami, S. Cooper, and D. Koller, "Programming pluralism: Using learning analytics to detect patterns in the learning of computer programming," *Journal of the Learning Sciences*, vol. 23, no. 4, pp. 561–599, 2014.

[7] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," in *Proceedings of the Fifteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 91–99.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[9] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," Royal Signals and Radar Establishment Malvern, United Kingdom, Tech. Rep., 1988.

[10] B. M. Brown, "Statistical uses of the spatial median," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 45, no. 1, pp. 25–30, 1983.

[11] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974.

[12] V. E. Castro and J. Yang, "A fast and robust general purpose clustering algorithms," in *Proceedings of the International Conference on Artificial Intelligence*, 2000.

[13] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Systems with Applications*, vol. 40, no. 1, pp. 200–210, 2013.

[14] C. Chen, K. Li, M. Duan, and K. Li, "Extreme learning machine and its applications in big data processing," in *Big data analytics for sensor-network collected intelligence*. Elsevier, 2017, pp. 117–150.

[15] S. D. Connell and A. K. Jain, "Learning prototypes for online handwritten digits," in *Proceedings. Fourteenth International Conference on Pattern Recognition*, vol. 1, 1998, pp. 182–184.

[16] G. Conole, D. Gašević, P. Long, and G. Siemens, "Message from the LAK 2011 general & program chairs," in *International Learning Analytics & Knowledge Conference 2011*. Association for Computing Machinery (ACM), 2011.

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[18] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[19] F. Dellaert, "The expectation maximization algorithm," College of Computing, Georgia Institute of Technology, 2003.

[20] A. P. Dempsterm, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[21] B. Desgraupes, "Clustercrit: Clustering indices," https://cran.r-project.org/web/packages/clusterCrit/clusterCrit.pdf, Accessed: 1 April 2022.

[22] J. K. Dixon, "Pattern recognition with partly missing data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 617–621, 1979.

[23] C. Dorai and A. K. Jain, "Shape spectra based view grouping for free-form objects," in *Proceedings of the International Conference on Image Processing*, vol. 3, 1995, pp. 340–343.

[24] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[25] E. Eirola, G. Doquire, M. Verleysen, and A. Lendasse, "Distance estimation in numerical data sets with missing values," *Information Sciences*, vol. 240, pp. 115–128, 2013.

[26] R. Engels and C. Theusinger, "Using a data metric for preprocessing advice for data mining applications," in *Proceedings of the European Conference on Artificial Intelligence (ECAI-98)*. John Wiley & Sons, 1998, pp. 430–434.

[27] V. Estivill-Castro, “Why so many clustering algorithms: A position paper,” *SIGKDD Explorations Newsletter*, vol. 4, no. 1, pp. 65–75, 2002.

[28] A. Farhangfar, L. Kurgan, and J. Dy, “Impact of imputation of missing values on classification error for discrete data,” *Pattern Recognition*, vol. 41, no. 12, pp. 3692–3705, 2008.

[29] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, pp. 37–37, 1996.

[30] E. W. Forgy, “Cluster analysis of multivariate data: efficiency versus interpretability of classifications,” *biometrics*, vol. 21, pp. 768–769, 1965.

[31] C. Fraley and A. E. Raftery, “How many clusters? which clustering method? answers via model-based cluster analysis,” *The Computer Journal*, vol. 41, no. 8, pp. 578–588, 1998.

[32] P. Fränti and S. Sieranoja, “K-means properties on six clustering benchmark datasets,” *Applied Intelligence*, vol. 48, no. 12, pp. 4743–4759, 2018. [Online]. Available: http://cs.uef.fi/sipu/datasets/

[33] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. USA: Kluwer Academic Publishers, 1991.

[34] T. F. Gonzalez, “Clustering to minimize the maximum intercluster distance,” *Theoretical computer science*, vol. 38, pp. 293–306, 1985.

[35] F. Gutierrez, “Cloud and big data,” in *Spring Cloud Data Flow*. Berkeley, CA: Apress, 2021.

[36] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “Cluster validity methods: part I,” *ACM Sigmod Record*, vol. 31, no. 2, pp. 40–45, 2002.

[37] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2011.

[38] P. E. Hart, D. G. Stork, and R. O. Duda, *Pattern classification*. Wiley Hoboken, 2000.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2009.

[40] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. USA: Prentice Hall, 1999.

[41] J. Hämäläinen, A. S. C. Alencar, T. Kärkkäinen, C. L. C. Mattos, A. H. Souza Junior, and J. P. P. Gomes, “Minimal learning machine: Theoretical results and clustering-based reference point selection,” *Journal of Machine Learning Research*, vol. 21, 2020.

[42] J. Hämäläinen, T. Kärkkäinen, and T. Rossi, "Improving scalable k-means++," *Algorithms*, vol. 14, no. 1, p. 6, 2020.

[43] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[44] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. USA: Prentice-Hall, Inc., 1988.

[45] S. Jauhiainen and T. Kärkkäinen, "A simple cluster validation index with maximal coverage," in *Proceedings of the 25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. ESANN, 2017, pp. 293–298.

[46] I. Jolliffe, "Principal component analysis," in *International Encyclopedia of Statistical Science*. Berlin: Springer, 2011, pp. 1094–1096.

[47] I. Katsavounidis, C. C. J. Kuo, and Z. Zhang, "A new initialization technique for generalized lloyd iteration," *IEEE Signal processing letters*, vol. 1, no. 10, pp. 144–146, 1994.

[48] L. Kaufman and P. Rousseeuw, "Finding groups in data: An introduction to cluster analysis," *Wiley, New York*, 1990.

[49] M. Kim and R. S. Ramakrishna, "New indices for cluster validity assessment," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2353–2363, 2005.

[50] T. Kärkkäinen and E. Heikkola, "Robust formulations for training multilayer perceptrons," *Neural Computation*, vol. 16, no. 4, pp. 837–862, 2004.

[51] T. Kärkkäinen and J. Toivanen, "Building blocks for odd-even multigrid with applications to reduced systems," *Journal of Computational and Applied Mathematics*, vol. 131, pp. 15–33, 2001.

[52] T. Kärkkäinen and S. Äyrämö, "On computation of spatial median for robust data mining," in *Proceedings of Sixth Conference on Evolutionary and Deterministic Methods for Design, Optimisation and Control with Applications to Industrial and Societal Problems, EUROGEN*, 2005.

[53] D. Lam and D. C. Wunsch, "Clustering, academic press library in signal processing," *Theory Machine Learning*, vol. 1, pp. 1115–1149, 2014.

[54] D. Laney, "3D data management: Controlling data volume, velocity, and variety," *META Group*, 2001.

[55] N. Li, W. W. Cohen, and K. Koedinger, "Discovering student models with a clustering algorithm using problem content," in *Proceedings of the 6th International Conference on Educational Data Mining*, 2013, pp. 98–105.

[56] Q. Li, S. Yue, and M. Ding, "Volume and surface area-based cluster validity index," *IEEE Access*, vol. 8, pp. 24 170–24 181, 2020.

[57] R. J. Little and D. B. Rubin, *Statistical Analysis with Missing Data*. USA: John Wiley & Sons, 1986.

[58] R. J. A. Little, "A test of missing completely at random for multivariate data with missing values," *Journal of the American Statistical Association*, vol. 83, no. 404, pp. 1198–1202, 1988.

[59] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *2010 IEEE international conference on data mining*. IEEE, 2010, pp. 911–916.

[60] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Volume 1: Statistics*. Oakland, CA, USA, 1967, pp. 281–297.

[61] A. Majumdar and R. K. Ward, "Some empirical advances in matrix completion," *Signal Processing*, vol. 91, no. 5, pp. 1334–1338, 2011.

[62] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *The Journal of Machine Learning Research*, vol. 11, p. 2287–2322, 2010.

[63] A. Mead, "Review of the development of multidimensional scaling methods," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 41, no. 1, pp. 27–39, 1992.

[64] D. P. P. Mesquita, J. P. P. Gomes, A. H. Souza Junior, and J. S. Nobre, "Euclidean distance estimation in incomplete datasets," *Neurocomputing*, vol. 248, pp. 11–18, 2017.

[65] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, 1985.

[66] ——, "A study of standardization of variables in cluster analysis," *Journal of classification*, vol. 5, no. 2, pp. 181–204, 1988.

[67] F. Murtagh, "A survey of recent advances in hierarchical clustering algorithms," *Computer Journal*, vol. 26, no. 4, pp. 354–359, 1984.

[68] Y. T. Mustafa, V. A. Tolpekin, and A. Stein, "Application of the expectation maximization algorithm to estimate missing values in gaussian bayesian network modeling for forest growth," *IEEE transactions on geoscience and remote sensing*, vol. 50, no. 5, pp. 1821–1831, 2011.

[69] M. Nakagami, "The m-distribution-a general formula of intensity distribution of rapid fading," in *Statistical Methods in Radio Wave Propagation*, W. Hoffman, Ed. Pergamon, 1960, pp. 3–36.

[70] F. V. Nelwamondo, S. Mohamed, and T. Marwala, “Missing data: A comparison of neural network and expectation maximization techniques,” *Current Science*, vol. 93, no. 11, pp. 1514–1521, 2007.

[71] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern Recognition*, vol. 37, no. 3, pp. 487–501, 2004.

[72] D. Pyle, *Data Preparation for Data Mining*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[73] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[74] E. Rasmussen, “Clustering algorithms,” in *Information Retrieval: Data Structures and Algorithms*. USA: Prentice-Hall, Inc., 1992, pp. 419–442.

[75] S. Ray and R. Turi, “Determination of number of clusters in k-means clustering and application in colour image segmentation,” in *Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques*, 1999, pp. 137–143.

[76] L. Rokach, *Clustering methods, Data Mining and Knowledge Discovery Handbook*. Berlin, Heidelberg: Springer-Verlag, 2005.

[77] M. Saarela and T. Kärkkäinen, “Discovering gender-specific knowledge from finnish basic education using pisa scale indices,” in *Proc. of the 7th International Conference on Educational Data Mining*, 2014, pp. 60–67.

[78] V. Satopää, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a" kneedle" in a haystack: Detecting knee points in system behavior,” in *2011 31st international conference on distributed computing systems workshops*. IEEE, 2011, pp. 166–171.

[79] A. Shademan and M. A. Zia, “Adaptive vector quantization of mr images using online k-means algorithm,” in *Applications of Digital Image Processing XXIV*, A. G. Tescher, Ed., vol. 4472, International Society for Optics and Photonics. SPIE, 2001, pp. 463–470.

[80] D. Steinley, “Local optima in k-means clustering: what you don’t know may hurt you.” *Psychological methods*, vol. 8, no. 3, p. 294, 2003.

[81] R. E. Strauss, M. N. Atanassov, and J. A. De Oliveira, “Evaluation of the principal-component and expectation-maximization methods for estimating missing data in morphometric studies,” *Journal of Vertebrate Paleontology*, vol. 23, no. 2, pp. 284–296, 2003.

[82] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to data mining*. Addison-Wesley, 2005.

[83] S. Theodoridis, *Pattern recognition*. Oxford: Elsevier Books, 2003.

[84] M. Thomas and A. T. Joy, *Elements of information theory*. Wiley-Interscience, 2006.

[85] R. L. Thorndike, “Who belongs in the family,” *Psychometrika*, vol. 18, pp. 267–276, 1953.

[86] R. Urbanowicz and J. Moore, “Exstracs 2.0: Description and evaluation of a scalable learning classifier system,” *Evolutionary Intelligence*, vol. 8, p. 89, 2015.

[87] J. Van Hulse and T. M. Khoshgoftaar, “Incomplete-case nearest neighbor imputation in software measurement data,” *Information Sciences*, vol. 259, pp. 596–610, 2014.

[88] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” *Journal of Machine Learning Research*, vol. 11, no. 95, pp. 2837–2854, 2010.

[89] M. Worsley and P. Blikstein, “Analyzing engineering design through the lens of computation,” *Journal of Learning Analytics*, vol. 1, pp. 151–186, 2014.

[90] W. Xing, B. Wadholm, and S. Goggins, “Learning analytics in CSCL with a focus on assessment: An exploratory study of activity theory-informed cluster analysis,” in *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge*, ser. LAK ’14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 59–67.

[91] R. Xu and D. Wunsch, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[92] ——, “Clustering algorithms in biomedical research: A review,” *IEEE Reviews Biomedical Engineering*, vol. 3, pp. 120–154, 2010.

[93] C. Zahn, “Graph-theoretical methods for detecting and describing gestalt clusters,” *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 68–86, 1971.

[94] Q. Zhao and P. Fränti, “Wb-index: A sum-of-squares based index for cluster validity,” *Data & Knowledge Engineering*, vol. 92, pp. 77–89, 2014.

[95] Q. Zhao, M. Xu, and P. Fränti, “Knee point detection on bayesian information criterion,” in *2008 20th ieee international conference on tools with artificial intelligence*, vol. 2. IEEE, 2008, pp. 431–438.

[96] S. Äyrämö, T. Kärkkäinen, and K. Majava, “Robust refinement of initial prototypes for partitioning-based clustering algorithms,” in *Recent Advances in Stochastic Modeling and Data Analysis*. World Scientific, 2007, pp. 473–482.

ORIGINAL PAPERS

I

GAME LEARNING ANALYTICS FOR UNDERSTANDING READING SKILLS IN TRANSPARENT WRITING SYSTEM

by

Marko Niemelä, Tommi Kärkkäinen, Sami Äyrämö, Miia Ronimus, Ulla Richardson, and Heikki Lyytinen 2020

British Journal of Educational Technology, 51(6):2376–2390

https://doi.org/10.1111/bjet.12916

Game learning analytics for understanding reading skills in transparent writing system

Marko Niemelä, Tommi Kärkkäinen, Sami Äyrämö, Miia Ronimus, Ulla Richardson and Heikki Lyytinen

Marko Niemelä is a PhD student at the Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland. His research interests include machine learning, data analytics, and optimization. Tommi Kärkkäinen is a professor at the Faculty of Information Technology, University of Jyväskylä. His main research fields include computational sciences and learning analytics. Sami Äyrämö is an adjunct professor of data analytics at the Faculty of Information Technology, University of Jyväskylä. His research interests include machine learning and predictive modelling with applications in sport, health and medicine. Miia Ronimus is a postdoctoral researcher at the Niilo Mäki Institute, Jyväskylä, Finland. Her research interests include digital game -based learning, student motivation, and dyslexia. Ulla Richardson is a professor at the Centre for Applied Language Studies, University of Jyväskylä. Her research interests include technology enhanced language learning, reading development, dyslexia, and reading skill assessment. Heikki Lyytinen is a professor at the Department of Psychology, University of Jyväskylä. He has UNESCO Chair on Inclusive Literacy Learning for All. His areas of recent research include dyslexia, reading acquisition, and digital learning environments. Address for correspondence: Mr. Marko Niemelä, Faculty of Information Technology, University of Jyväskylä, PO Box 35, FI-40014 Jyväskylä, Finland. Email: marko.p.niemela@jyu.fi

Abstract

Serious games are designed to improve learning instead of providing only entertainment. Serious games analytics can be used for understanding and enhancing the quality of learning with serious games. One challenge in developing computerized support for learning is that learning of skills varies between players. Appropriate algorithms are needed for analyzing the performance of individual players. This paper presents a novel clustering-based profiling method for analyzing serious games learners. *GraphoLearn*, a game for training connections between speech sounds and letters, serves as the game-based learning environment. The proposed clustering method was designed to group the learners into profiles based on game log data.

The obtained profiles were statistically analyzed. For instance, the results revealed one profile consisting of 136 players who had difficulties with connecting most of the target sounds and letters, whereas learners in the other profiles typically had difficulties with specific sound-letter pairs. The results suggest that this profiling method can be useful for identifying children with a risk of reading disability and the proposed approach is a promising new method for analyzing serious game log data.

Keywords: learning analytics, serious game, letter knowledge, reading difficulties

Introduction

Background and motivation

Differentiated instruction is a framework supporting diverse needs and ability levels of students in classrooms by flexible use of time, space, materials, and strategies (Regan et al., 2014). Computer-assisted instruction, including intelligent tutoring systems and serious games is one way to differentiate traditional teaching (Boone & Higgins, 2007). Intelligent tutoring systems usually focus on embodying learning principles and adapting for differences between students, where as serious games emphasize student's motivation and engagement (Yanjin & Vincent, 2017). Serious games provide a considerable alternative for improving learning experience in comparison to traditional teaching methods such as classroom lessons (Wendel et al., 2012). Many serious games share the features of intelligent tutoring systems by including individually adaptive learning content, and by logging game events and contextual information during the gameplay (Wendel et al., 2012). Adaptation usually includes automatic content creation and adaptation of difficulty level for individual users as well as adaptation rules for gameplay (Wendel et al., 2012). Therefore, serious games provide an excellent platform for collecting data about individual differences in learning, which can be analyzed and utilized in the development of differentiated instruction.

Practitioner Notes

What is already known about this topic

- Serious games are used to improve learning and to tailor learning environments for people with various difficulties in learning.
- Learning analytics and serious games analytics are growing research fields, applying and developing data analysis methods to analyze, profile, and understand learning using serious games.
- *GraphoLearn* is a learning game for training reading skills. The game provides preventive support for learners with varying skill levels including individuals who are struggling with reading.

What this paper adds

- The paper develops and presents a novel approach for serious games analytics to analyze *GraphoLearn* players.
- The proposed data analysis approach produces an interpretable set of error profiles, which characterize the learning difficulties in a unique way.
- The profiling method can be used for longitudinal studies and applied to analyzing logs of other serious games.

Implications for practice and/or policy

- It is possible to reveal and understand profiles of serious game players.

- The proposed data analysis method can be used to identify players who have a potential risk for reading difficulties or disabilities.
- Even though the proposed method provides only limited information about players' future skills, it offers a good starting point for other studies in which players' development can be monitored more accurately.

Learning analytics focuses on the development and utilization of analysis methods for data from educational settings. The main ambition of learning analytics is to measure, collect, analyze, and report data about learners, for purposes of understanding and optimizing learning, teaching, and the environment in which it all occurs (Mor et al., 2015). It aims for the discovery of meaningful patterns about learners in their learning environment by using methods originated from statistics, information visualization, data mining, and social network analysis (Chatti et al., 2012; Peña-Ayala, 2017). Learning analytics can respond to a wide range of different needs, including visualization of learning activities, assessing learning behavior, predicting student performance, learning personalization, profiling, evaluation of social learning, and improving learning materials and tools (Nguyen et al., 2017).

In the present study serious games analytics is applied to the Finnish version of *GraphoLearn*. The game was originally developed during the Jyväskylä Longitudinal Study of Dyslexia (JLD) (Lyytinen et al., 2009). The aim was to support the basic decoding skills of Finnish children at risk for reading difficulties by helping the learner to connect spoken items (e.g. speech sounds) to their written counterparts (e.g. letters). Nowadays, the game has been adapted accordingly to a high variety of languages around the world.

We combine the methods of clustering, missing values handling, and cluster validation to offer an approach for profiling *GraphoLearn* players. The proposed model categorizes learners into distinct profiles based on players' game log data informing about the choices the players have made in the game. The number of profiles is selected by using cluster validation indices. We excluded other more complex clustering methods because we do not aim at discovering clusters with any specific or anomalous shapes, but rather partition the data into subsets of similar observations using a clustering model that is straightforward to interpret both with respect to input variables and players (Steinbach et al., 2004). Further, the study presents statistics of the different profiles, which can be used for analyzing learners' risk for a reading difficulty. The purpose of the research is to identify a distinct set of learner profiles, which are interpretable and applicable to practice.

On serious games analytics

Serious games analytics can be used to improve learning and to tailor learning environments for people with various difficulties in learning. Lameras et al. (2017) investigated how learning attributes (e.g., learning activities, learning outcomes, assessment, and feedback) and game properties can be planned, designed, and implemented by university teachers interested in using games for teaching and learning in higher education. The study identified 165 papers providing empirical evidence and conceptual assumptions concerning specific learning activities that could be linked with game elements (e.g., leaderboard, virtual currencies, and in-game hints), feedback and progress indicators, and teacher's roles designing and facilitating game play. Nguyen et al. (2018) provided a framework and a design tool for people with intellectual disabilities to address each learner's individual needs. The proposed framework is valuable for the design, implementation, evaluation, and adaptation of serious games for more enhanced learning and teaching at the group or individual level

Serious games analytics can also be successfully applied for analyzing the individual differences and behavioral patterns of serious game learners. For instance, Hicks, Eagle, et al. (2016) analyzed gameplay patterns of the *Quantum Spectre* physics game to understand player dropout in the game. By using survival analysis, interaction network analysis, and the results from player surveys they were able to identify particular problem spots where players dropped out of the game due to its complexity. Hicks, Liu, Eagle, and Barnes (2016) also compared three different level creation editors, which are helpful for players learning about the *BOTS* game's core mechanic. Based on the results of a zero-inflation model, programming editor and building editor were more effective than drag-and-drop editor in the case of encouraging the creation of levels, which contained more game play affordances for players. Horn et al. (2016) explored player strategies in *GrAZE*, an educational puzzle-based game that is designed to support algorithmical thinking for middle school students. The aim was to understand by using hierarchical clustering how players learn and progress in the game. The study identified problem areas in the game design for further development of the game. Harpstead and Aleven (2015) utilized learning curve analysis from serious games analytics in *BeanStalk* physics game designed to teach the concept of balance beam system for young children. The aim was to find implications for the level design to better accomplish its educational goals. The results show that analytical methods can yield actionable design recommendations.

Research questions

The study uses learning analytics for analyzing the playing patterns of *GraphoLearn* players based on a group-level information extracted from cluster profiles. The variables of interest are error rates, contexts of the errors, progression information, total playing times, and interval times between playing sessions. The provided clustering method is a novel alternative for analyzing partially incomplete learning data and it is modifiable for a high volume of data. The developed method is aimed to help characterizing and monitoring players and their learning process. In addition, the method can help researchers identify groups of individuals who have a risk of reading difficulty. A diverse set of profiles is expected to be found because of a relatively large sample of different learners. The research questions are following:

RQ1: Is it possible to identify a set of distinct and interpretable cluster profiles by using the proposed clustering method?

RQ1.1: Can internal cluster validation indices be used for finding the number of clusters in the models that are well-separated, interpretable, and useful for practice.

RQ2: What are the typical bottlenecks compromising the learning of letter-sound correspondences?

We set the following hypotheses to the research questions:

H1: Because of the variability in the starting skills of the learners, we expect to identify several distinct profiles which can be interpreted for further application.

H1.1: We expect that the use of cluster validation indices lead to a number of clusters that are well-separated, interpretable and thereby useful as well (see e.g. Hämäläinen et al., 2018).

H2: We anticipate children to confuse especially letters that either look or sound similar (see Lyytinen et al., 2009).

Context of the study

Reading skill development

The basic reading skill is based on connection building between spoken and written language. Thus, learning the skill requires storing of those connections. *GraphoLearn* is designed as a training environment for this purpose (Richardson & Lyytinen, 2014). In alphabetic writing systems, such connection building is based on the smallest imaginable units, phonemes, and their written equivalents, that is, letters (or graphemes when more than one letters is used to represent one sound). Phonemes and graphemes are consistently connected in transparent orthographies. Thus, one has to learn only the sounds of letters and invent that assembling such sounds in the order of letters means reading. In less transparent writings systems such as English the same principle works but only by using larger units such as rimes (e.g. *ing* in English) to make the connections more "learnable", that is, true in all contexts of writing. Learning to differentiate phonetically similar sounds such as */g/* and */d/* and visually similar letters such as *n* and *h* can be considered as the most challenging part of storing the connections. The method described here helps to understand reading difficulties and disorders, which result from, for example, biological factors or inadequate education. This is made by showing how the difficulties appear during the learning (i.e., connection building) process.

GraphoLearn

GraphoLearn is a game proven to provide preventive support for learning to read (Saine et al., 2011). The game was originally developed as a way to observe how the difficulties in learning appear and later to supplement for reading instruction provided by schools. There are dozens of different *GraphoLearn* versions built for helping the learner to master the connection building in different linguistic and orthographic contexts. The present study used the version designed for Finnish students.

In transparent orthographies the game starts by introducing speech sounds and corresponding letters. First, phonetically and visually distinct and easy to perceive letters (e.g., *a*, *e*, and *i*) are presented and then one moves on to present correspondences that are more similar and thus less distinguishable (e.g., *b*, *d*, and *p*). In the game, the player first hears speech sound and then identifies and selects the corresponding letter from the several alternatives shown on the screen. The player receives immediate visual and auditory (corrective) feedback after each response. When the player has learned to connect most of the sounds and letters flawlessly, the game proceeds to training larger units such as spoken and written syllables and then words, starting from two letter syllables and eventually moving on to long words consisting of several letters. The player is expected to grasp the idea that reading occurs by assembling the speech units represented by the letters of a word.

An important feature of *GraphoLearn* is that the progression of the game adapts to the learner's current level of performance. This is done, for example, by using the Bayesian principle to present new learning tasks (Kujala et al., 2010). The adaptation techniques aim for a mean success rate of at least 80%, offering both challenge and success, which together makes playing more rewarding. Important features are also a personalizable avatar and rewards. Such rewards and graphically different game levels are efficient ways to sustain the learners' motivation in playing and to expose them repeatedly to strengthen the correct connections. The game also involves the static assessment levels of learners' development in the tasks during playing.

Figure 1 shows the user interface of an assessment task included in specific versions of *GraphoLearn*, and chosen for a closer inspection in the present study. The assessment task evaluates the player's skill in identifying the letters corresponding to the 23 speech sounds of the Finnish language. In the assessment, the player hears each of the sounds, one by one, and selects the corresponding letter from the alternatives shown on the screen. The sound is repeated if player does not response within 5 seconds. If the player does not answer within 15 seconds, an option for skipping the trial becomes available. The assessment is first presented when the game is started and is then repeated at intervals of 1 hour.

Methods

Participants

Learners were recruited by sending an information letter about *GraphoLearn* and upcoming study to an email list of teachers registered as *GraphoLearn* users. The information letter was sent in September, about six weeks after the start of the school year. Teachers were asked to consider if they had a first grade student with risk factors for dyslexia (difficulties at learning to read, poor letter knowledge, family members with dyslexia) and who spoke Finnish as first language. *GraphoLearn* was recommended for such students. Teachers needed a written consent from the child's guardian before registration. Before the game could be used, parents and teachers also

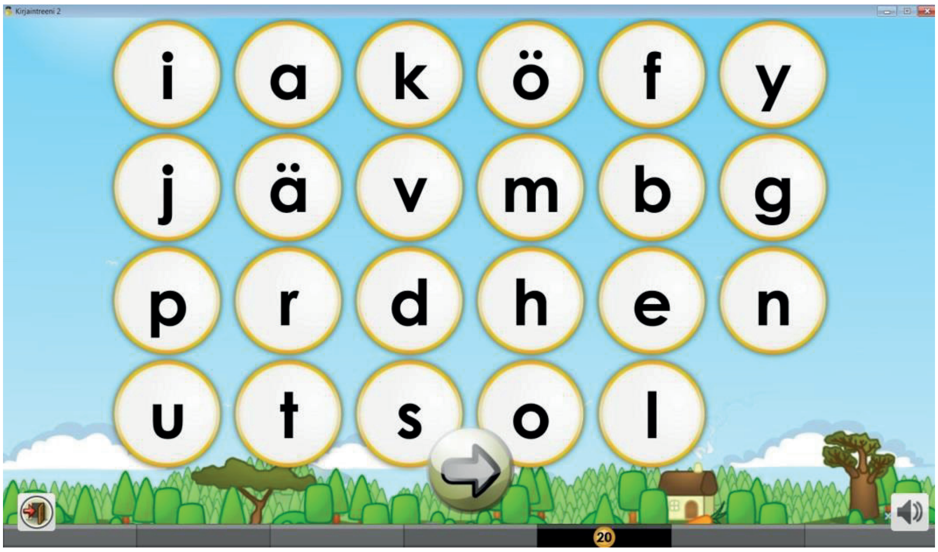


Figure 1: Appearance of the sound-letter assessment task included in GraphoLearn

needed to accept the terms and conditions stating that the game log data would be saved in the secure *GraphoLearn* server and could be used for research purposes

Although we were unable to control the type of children who started the use the game, we expect that the suggestions given in the information letter had an effect on the characteristics of the sample, and we expect it to consist mostly of Finnish-speaking first graders who had not yet acquired the level of letter-sound correspondence skills needed for learning to read and who may also have a risk of dyslexia.

Eventually, data was gathered from 1632 players who were 6.5–8.75 years old (*M=7.39, SD=0.46*). The data from the sound-letter assessment task indicates that children could correctly identify 13.68 letters (*SD*=5.09) out of 23, suggesting they had not yet learnt to master all the associations between sounds and letters and would likely benefit from training with *GraphoLearn.*

Majority of the players were boys (61.1%), which is probably because reading difficulties are more common among boys (e.g. Rutter et al., 2004). The players came from more than 200 municipalities with all regions of Finland being represented. Largest numbers of players came from the cities of Helsinki and Jyväskylä. The number of adults, who had registered the children as *GrahoLearn* players, was 669. These adults, 88.6% being teachers and 11.4% parents, were in charge of supervising the player. The number of registered players per adult ranged from 1 to 46, but only 3% were in charge of more than 10 players. The median number of registered players per adult was 1.

Data collection

The players can learn to use *GraphoLearn* within 1–2 minutes. They were advised to use headphones and play short (about 10 minutes) sessions at time, and several sessions per day in consecutive days. Teachers and parents were responsible of supervising the playing and ensuring that children used the game in a quiet place to avoid distractions. Teachers and parents were advised not to help children with game tasks, so that the difficulty level determined by the adaptation would not increase too much relative to child's skill level.

The player's actions during the game were logged into a database. The personal log files include, for example, starting times, ending times, number of playing sessions, target items, durations, correct/incorrect selections, and skipped tasks. For research purposes, the most important information to be logged were player's inputs and time spent with each task (from perceiving the stimuli to the selection of the corresponding written unit), which is commonly referred as response time. The sample was divided into two groups based on the type of letters used in practice (lowercase or uppercase) that was chosen by user. In total 1275 players played with lowercase letters and 357 players used uppercase letters. The lowercase letters are used in the initial stages of formal reading instruction at schools, which is the likely reason for them being chosen more often. The main limitation of the data was that 4.66% of the responses were missing because some players stopped playing before all 23 targets had been presented. This was taken in account when algorithms were developed for the present analysis (see the next section).

The realized profiling approach

Clustering is an unsupervised technique for organizing empirical observations into different groups called clusters so that observations in the same cluster are more similar to each other than observations in the other clusters. K-means is probably the most common prototype-based partitional clustering approach, which has a long history (Jain, 2010). The algorithm is broadly used due to its ability to solve general purpose problems. K-means finds a partition such that the squared Euclidean error between cluster prototype, and the observations in the cluster is minimized (see more details in Supplement S2).

Many clustering algorithms require the number of clusters as an input parameter. However, this information is not often available and it can be a challenging task to determine the number, especially in the cases of multidimensional data. Even though there exist different tricks to illustrate multidimensional data, for example, using different multidimensional visualization techniques or dimension reduction techniques, perceiving the data structure may not be obvious. Cluster validity measures provide a way of validating the quality of results of clustering methods to find a partition that best fits the nature of data. Because of multidimensional data structures, cluster validation measures, for example, cluster validation indices, are very suitable, even essential methods, for determining the number of clusters (Arbelaitz et al., 2013). The internal cluster validity index is one of categories of cluster validity, which utilizes the results of a clustering algorithm in terms of quantities of the data set itself (see more details in Supplement S3).

This study consisted of implementing K-means clustering, K-means++ initialization, and cluster validation indices algorithms. Since some observations in *GraphoLearn* data included missing values, distance calculations in all of the implemented algorithms was needed to replace with the general similarity measure (Gower, 1971).

The players were divided into two groups based on whether they used lowercase or uppercase letters. The game log data of both groups were transformed into two binary matrices. The number of rows corresponded to the number of players and the number of columns to the number of distinct target-letter pairs. Non-zeros in the matrices indicated selected erroneous selections. Matrix dimensionalities were reduced, because of the computational cost of clustering. This was applied by filtering out columns, which did not consisted noticeable number of erroneous selections.

After the pre-processing step, the clustering was performed by gradually increasing the number of clusters, *K*, from 2 to 10. The maximum number was selected as 10, because a high number of clusters makes the interpretation and analysis of the results more challenging. In addition, a small number of *K* generalizes data the most. For instance, Saarela and Kärkkäinen (2015) used 11 as the maximum number of clusters in their study of the Finnish student population in PISA 2012. For each value of *K*, clustering was repeated 200 times and the best prototypes with the lowest clustering error were saved. These were also used as initial points for the next value of *K*, where the additional initial point was generated using K-means++ initialization algorithm.

The quality of distinct data partitions and obtained cluster profiles were evaluated using internal cluster validation indices (CVIs). Eight CVIs were selected from our previous study (Niemelä et al., 2018) for calculating clustering index values. Multiple indices were selected to the current study since the previous studies revealed that there does not exist one superior index which overcomes others (see e.g. Hämäläinen et al., 2017). Each CVI produced one quality measure of clustering for each value of *K*. These values were used when deciding the final number of clusters for lowercase and uppercase data sets. Index values from different indices were scaled to the same range of [0, 1] to easy up their comparison.

The number of clusters was decided by analyzing the index curves of validation indices. First, the index values were grouped together and the speed of improvement (i.e., strength of decreasing trend based on group distributions) was analyzed using statistical testing. The aim was to reject weak candidates, that is, to eliminate regions where improvements were not statistically significant. The Wilcoxon statistical ranksum test was performed for each two-pair of successive groups. In the final stage, the number of clusters were decided benefiting the statistical measures and analyzing figures obtained from the CVIs. Regarding the source codes of algorithms, they are available online[1].

Results

Interpretation of the learner profiles

Figures S1.1, S1.2, and S1.3 in Supplement S1 show the learner profiles in a confusion

[1] http://users.jyu.fi/~mapeniem/BJET/Kmeans/

matrix format for the lowercase letter data set. The profiles were calculated also by using the uppercase letter data set but because of similar confusion patterns and low number of cases in certain profiles they are omitted here. Nevertheless, these results are available in Supplement S3.

In Figures S1.1, S1.2, and S1.3 darker colors indicate higher average confusion percentages for the target-distractor pairs over the players in the profiles. The confusions in the matrix diagonals are zero because they indicate correct selections. Most of the observed confusion can be explained by phonetic and visual similarity of the sounds and letters. These two main categories of confusion are marked with "circle" and "square" symbols in the matrices. It is also possible that the errors are associated to both or neither categories, which are marked with "star" and "rectangle" symbols.

Main errors in the profiles

Confusion symbols are summarized in Table 1. Only confusion percentages exceeding 10% are illustrated to clarify presentation. Further, noticeable confusions exceeding 15% are underlined. Table 1 shows that many profiles have something in common, for example, the letter *n* is often mixed to letters *h*, *m*, and the letter *f* is mixed to letters *s* and *v*. Especially, *n* is strongly confused with *m* and this can be concluded to be the most challenging sound-letter pair for the players possibly, because both acoustic and visual similarity compromises building the connection. An interesting finding is that the confusion between commonly mixed letters *f* and *v* cannot be explained by concrete phonetic nor visual similarity of the letters. This may be related to *f* being a foreign letter in the Finnish language, and being pronounced as */v/* in certain dialects.

Table 1 shows that all the profiles have some unique errors regarding target letters. *Profile 1* players have difficulties in connection building due to the difficulties in separating both visually and phonetically similar items represented by the *b* and *d* letters, which is not as often appearing in other profiles. *Profile 2* players mix sound */g/* to sounds */d/* and */k/*, whereas *profile 3* players mix sound */t/* to sound */s/*. *Profile 4* players have difficulties with both of the two main confusion categories, that is, they often do not differentiate visually and phonetically similar letters. The main problems of *profile 5* and *profile 6* players are related to the visual similarity of the letters.

Calculated statistics

Table 2 provides information about the performances in the assessment tasks and playing patterns of the players in the different profiles. The error rate refers to the mean percentage of incorrect selections of players within a profile. The players' development in connecting speech sounds to letters from the first assessment to the second assessment (after about 60 minutes of playing) was calculated by subtracting the error rate in the second assessment from the error rate in the first assessment. Only players who completed both assessments were included and clustering was not repeated in the second assessment. The total playing time refers to the time the game was used within the first five months of usage. The interval time refers to the median time gap between play sessions during the first month of playing.

Table 1: Symbol table of similarities for different profiles

	target | distractor																							
	b|d	b|p	b|v	d|b	d|g	f|h	f|s	f|v	g|b	g|d	g|k	i|l	j|l	m|n	n|h	n|m	p|b	p|d	t|f	t|s	u|o	y|ö	ö|o	ö|ä
profile																								
p1	☆			☆		○		▭	○	○		□		☆		☆		☆				○		
p2		☆			○		○	▭		○	○	□			□	☆	☆		□			○		□
p3							○	▭		○					□	☆	☆			○				
p4			○				○	▭		○		□			□	☆	☆				○	○	□	
p5		☆		☆				▭	○	○		□			□	☆	☆		□					
p6												□	▭	☆	□				□					
total	1	2	1	2	1	1	3	5	2	5	1	5	1	2	5	5	4	1	3	1	1	3	1	1

phonetic similarity=○, visual similarity=□, phonetic and visual similarity=☆,
unknown category=▭

According to Table 2, majority of players (34.8%) were grouped in the *profile 3*. In this profile, all the statistical values were near the average values of all profiles. The players in the *profile 4* have average error rate of 55.3% and median total playing time of 130.1 minutes. These values are much higher than the values in the other profiles. The *profile 4* players seem to have had difficulties with almost all target letters. The players have approximately 71% higher total playing time compared to the median value of all players, suggesting that they have needed more training than others. The average error rate in the *profile 6* is also high but this is caused by players who skipped most of the target tasks. The high percentage of the skipped tasks may imply that the players of this profile (5.4% of all players) were not motivated to complete the assessment in the beginning of the training. Although the players in the *profile 4* and *profile 6* had the highest error rates in the beginning, they also showed more progress than the players in other profiles according to the calculated differences in error rates, 25.1% and 23.9%, respectively. The players in *profile 1* and *profile 4* had the shortest time intervals between the playing sessions, suggesting more frequent playing.

Determination of the number of profiles

Using validation index curves obtained from different CVIs, minimums correspond to the best clustering structures. However, instead of the minimums, the speed of improvements of the index values was the main interest. Thus, because if the value of an individual CVI does not change much, it usually means that increasing the number of clusters does not notably improve the final solution. The results of CVIs are given in Figure 2. Numbers of clusters are in x-axes and y-axes show index values which were scaled to the range of [0, 1]. All indices except *Pakhira-Bandyopadhyay-Maulik (PBM)* and *Silhouette* obtain the minimum at the highest *K* value. Especially, *Calinski-Harabasz*, *kCE*, *PBM*, and *WB* indices provided the high speed of improvement of the cluster validation measures.

Table 2: Findings of profiles

	p1	p2	p3	p4	p5	p6	p
statistics							
size (in %)	14.3%	19.0%	34.8%	10.7%	15.8%	5.4%	100.0%
error rate	35.7%	39.1%	40.1%	55.3%	30.7%	54.2%	40.2%
progression*	14.3%	17.3%	17.6%	25.1%	11.5%	23.9%	17.8%
playing time	67.2 min	89.7 min	75.4 min	130.1 min	59.8 min	75.7 min	76.0 min
interval time	3.0 days	6.5 days	5.0 days	4.0 days	5.5 days	6.5 days	5.1 days

*Only players who completed both assessments are included.

Figure 3 shows a box plot presentation of all index values combined in the groups based on values of *K*. On each box the central mark indicates median of eight indices, and the bottom and the top edges of the box indicate 25th and 75th percentiles, respectively. The whiskers show to the most extreme data points and outliers are plotted using a '+' symbol. Table 3 presents statistical differences between each two pairs of groups, which were measured by Wilcoxon ranksum test so that only the successive groups which showed a decreasing trend in index values were compared. The bolded numbers indicate statistically significant differences between groups (*p<0.05*). Using the measured index values for lowercase letter data, statistically significant differences were obtained in two comparisons of the distributions. The measured difference between the median values of groups 5 and 6 was the highest (0.464) and therefore *K*=6 was the selected number of cluster profiles. Analogously, using the measured values for uppercase letter data, in total two comparisons were statistically different. The calculated difference between the median values of groups 2 and 5 was the highest (0.361) and therefore *K*=5 was the selected number. Nevertheless, our experiments showed that the fifth uppercase letter cluster profile included only few players and therefore four profiles were considered in the future analysis.

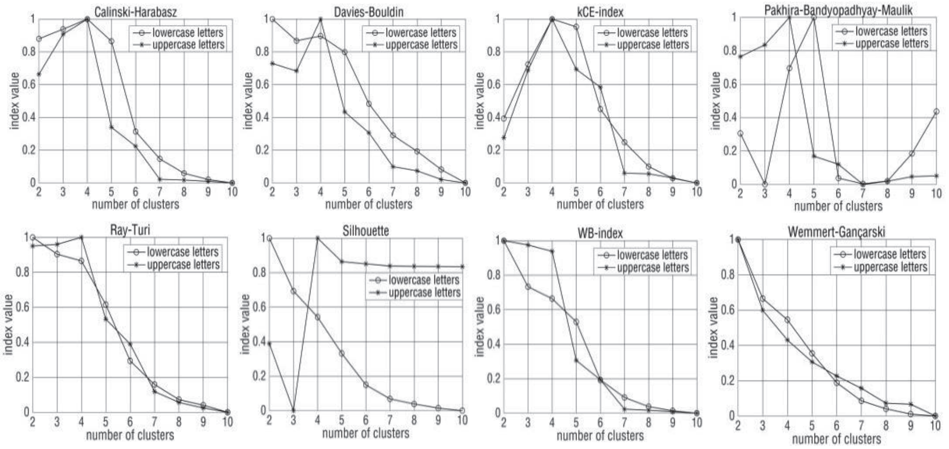


Figure 2: Values of cluster validation indices for K=2,…,10

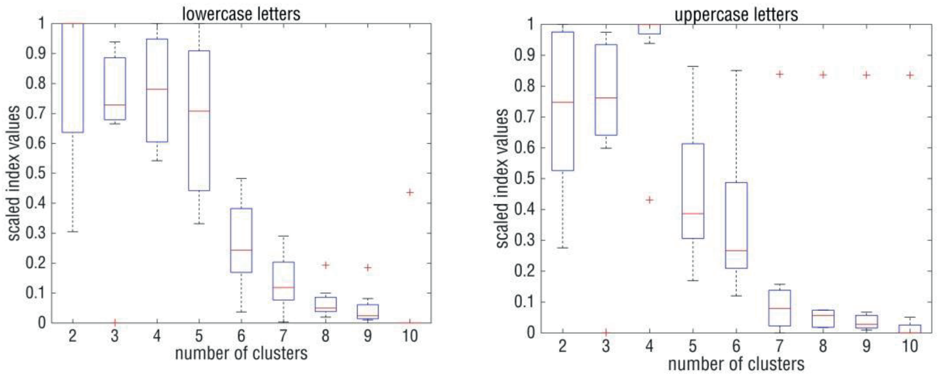


Figure 3: Box plot presentation of scaled index values

Table 3: Statistical p values obtained by Wilcoxon ranksum test

	compared pairs of distributions							
	g2, g3	g2, g5	g3, g5	g5, g6	g6, g7	g7, g8	g8, g9	g9, g10
lowercase letters	0.088	–	0.850	**0.002**	0.054	0.140	0.162	**0.003**
uppercase letters	–	**0.038**	–	0.104	**0.004**	0.326	0.521	0.238

Discussion

This paper presents a new clustering based approach for identifying different profiles of serious game players. We applied this method to *GraphoLearn* game log data. Based on the results, a set of profiles with different error types and rates were found. Even though there were errors common to all profiles, there were also many specific errors, which differentiated the profiles. According to Table 2, there were one "high" performing, three "medium" performing, and two "low" performing profiles with the different sound-letter pair errors. The players in the two weakest profiles showed the best progression while playing the game, which suggest that the combination of *GraphoLearn* and school-provided reading instruction helps children who have difficulties in reading acquisition. These findings are applicable to the practice and, therefore, the first hypothesis H1 is supported.

We found support to the hypothesis H2, because most of the errors were related to confusing phonetically and visually similar letters (see Table 1 for more details). Taking into account the confusions exceeding 10%, we realized that only 6 cases out of 57 confusions were not explainable by phonetic or visual similarity of letters.

Lyytinen et al. (2009) believed that children with familiar risk of dyslexia and/or low letter knowledge during the few months before school entry benefit from preventive playing in terms of avoiding unwanted failure experiences during the first months of school instructions. The study shows that the most challenging game tasks are related to visually and phonetically similar letters. In addition, uncommon letters in the Finnish language (e.g. *d* and *b*) showed to be challenging for the beginners.

The hypothesis H1.1 was supported. We used the Wilcoxon's ranksum test and the real differences of combined groups of CVIs to identify the number of clusters for lowercase and uppercase letter data. The results revealed 6 profiles for lowercase data and 5 profiles for uppercase data and we consider them as the most appropriate number for the clustering models.

Clustering methods are very commonly used in learning analytics. Saarela and Kärkkäinen (2017) have made a small survey of educational clustering methods. Three main approaches were hierarchical clustering, K-means clustering, and expectation maximization. These methods were used student modelling which included behavior and performance based models. The set of papers was identified scanning through relevant publication forums including the *Journal of Learning Analytics* and the *Conference on Learning Analytics & Knowledge.*

The used K-means clustering method and provided data analysis differentiates the current study from the related work as described in the section *On serious games analytics*. Horn et al. (2016) used the hierarchical clustering method to analyze game progression of learners. The main difference between clustering approaches is that the K-means clustering produce a single-layer clustering structure whereas the hierarchical method generates a tree-type clustering structure. The computational complicity of the hierarchical method is much higher and, therefore, it is not recommended for large-sized data sets. Further, the hierarchical method produces arbitrary shaped clusters whereas K-means produces easily interpreted geometrically closed subsets (Jain, 2010).

In the present study, the game data is limited only to the assessment tasks. To obtain more accurate and reliable clustering results, a larger sample size should be used. Further, other interesting variables could also be clustered, for example, larger units such as syllables or words, to achieve player profiles revealing differences in the types of errors children make in the actual reading. Further, more efficient clustering algorithms are required for a larger pool of samples. More specifically, a parallel implementation of algorithms into multiple machines with shared memory resources could be realized (Hämäläinen et al., 2018). Since *GraphoLearn* data contain missing values and outliers it is important to consider use of a robust clustering method in future algorithm design. For instance, spatial median is a statistically robust location estimate in clustering which can handle up to 50% of missing values or outliers (Hämäläinen et al., 2017)

A possible direction for future research could be repeating the clustering at regular time intervals to see how players divide into profiles in the follow-up cluster models. The approach offers a way to monitor players' progression in the game by detecting their connections to varied skill profiles. This new framework can be beneficial for validating the design of the original game, for example, it might be advantageous to improve the adaptation mechanism of the *GraphoLearn* for learners from different profiles (Kujala et al., 2010). For instance, Cano et al. (2018) have previously used learning analytics for validating the design of a learning game for adults with intellectual disabilities. In the study, the data tracker sent out relevant information about the behavior of the users and their learning patterns while playing the game. Further, statistical learning models, for example, neural networks, can be used for predicting players' game progression. Interesting variables to be predicted are, for example, player's inputs to different tasks and a particular time when the player will stop playing.

Conclusions

The growth of learning games and e-learning platforms imply that volumes of data on learning and learners are increasing rapidly. This means that special techniques are needed for analyzing learners with varying skills and their needs to enhance their learning process. We applied the clustering method from a branch of learning analytics to analyze performance of *GraphoLearn* players. The results indicated that it is possible to identify different types of learners using the given clustering method. The calculated statistics offered valuable information about the cluster profiles. This information can be used, for example, as a support for tracking children with a risk of reading disability due to certain types of bottlenecks compromising learning. Clustering was performed for data obtained at a very early stage in the game. Therefore, the used approach gives limited evidence about players' future skills. However, the future research direction is to extend the developed algorithms so that many other interesting learner patterns can be extracted from the data, for example, players' development in the game is one main interest. The present study offered the method, which is a considerable alternative for analyzing learners of alphabetical learning games and it is a good starting point for developing more effective analytical tools in different contexts of learning.

Statements on open data, ethics and conflict of interest

The data that support the findings of this study are available on request from the first author. The data are not publicly available due to them containing information that could compromise privacy of the participants.

Data collection, analysis and publishing followed the modes of action endorsed by the research community: integrity, meticulousness and accuracy in conducting research. The research ethics guidelines of the Finnish Advisory Board on Research Integrity (2019) were followed throughout this study.

There is no conflict of interest.

Acknowledgements

This study was supported by a grant from the Academy of Finland for the Unesco professor Heikki Lyytinen (decision numbers 292493 and 311737), the professor Tommi Kärkkäinen (decision numbers 311877 and 315550), and the professor Ulla Richardson (decision number 274050). The study was also supported by a grant from the Foundation Botnar for the professor Ulla Richardson (decision number 6066). All the authors gratefully acknowledge the postdoctoral researcher Harri Ketamo for the valuable discussions related to serious games and data analytics.

References

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., PéRez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256.

Boone, R. & Higgins, K. (2007). The role of instructional design in assistive technology research and development. *Reading Research Quarterly*, 42:135–140.

Cano, A. R., Fernández-Manjón, B., & García-Tejedor, A. J. (2018). Using game learning analytics for validating the design of a learning game for adults with intellectual disabilities. *British Journal of Educational Technology*, 49:659–672.

Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. *International Joural of Technology Enhanced Learning*, 4(5- 6):318–331.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics* (pp. 857-871). Washington, United States: International Biometric Society.

Hämäläinen, J., Jauhiainen, S., & Kärkkäinen, T. (2017). Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 10(3):105.

Hämäläinen, J., Kärkkäinen, T., & Rossi, T. (2018). Scalable robust clustering method for large and sparse data. In *Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 449–454). Bruges, Belgium: ESANN.

Harpstead, E., & Aleven, V. (2015). Using empirical learning curve analysis to inform design in an educational game. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play* (pp. 197–207). New York, United States: Association for Computing Machinery.

Hicks, A., Liu, Z., Eagle, M., & Barnes, T. (2016). Measuring gameplay affordances of user-generated content in an educational game. In T. Barnes, M. Chi, & M. Feng (Eds.), *Educational Data Mining* (pp. 78–85). North Carolina, United States: International Educational Data Mining Society (IEDMS).

Hicks, D., Eagle, M., Rowe, E., Asbell-Clarke, J., Edwards, T., & Barnes, T. (2016). Using game analytics to evaluate puzzle design and level progression in a serious game. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge* (pp. 440–448). Edinburgh, United Kingdom: Association for Computing Machinery.

Horn, B., Hoover, A. K., Barnes, J., Folajimi, Y., Smith, G., & Harteveld, C. (2016). Opening the black box of play: Strategy analysis of an educational game. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play* (pp. 142–153). Austin, Texas, United States: Association for Computing Machinery.

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

Kujala, J., Richardson, U., & Lyytinen, H. (2010). A bayesian-optimal principle for child-friendly adaptation in learning games. *Journal of Mathematical Psychology*, 54:247–255.

Lameras, P., Arnab, S., Dunwell, I., Stewart, C., Clarke, S., & Petridis, P. (2017). Essential features of serious games design in higher education: linking learning attributes to game mechanics. *British Journal of Educational Technology*, 48(4):972–994.

Lyytinen, H., Erskine, J., Kujala, J., Ojanen, E., & Richardson, U. (2009). In search of a science-based application: a learning tool for reading acquisition. *Scandinavian Journal of Psychology*, 50:668–675.

Mor, Y., Ferguson, R., & Wasson, B. (2015). Editorial: Learning design, teacher inquiry into student learning and learning analytics: A call for action. *British Journal of Educational Technology*, 46:221–229.

Nguyen, A., Gardner, L. A., & Sheridan, D. (2017). A multi-layered taxonomy of learning analytics applications. In R. A. Alias, P. S. Ling, S. Bahri, P. Finnegan, & C. L. Sia (Eds.), *21st Pacific Asia Conference on Information Systems.* Langkawi, Malaysia: PACIS.

Nguyen, A., Gardner, L. A., & Sheridan, D. (2018). A framework for applying learning analytics in serious games for people with intellectual disabilities. *British Journal of Educational Technology*, 49(4):673–689.

Niemelä, M., Äyrämö, S., & Kärkkäinen, T. (2018). Comparison of cluster validation indices with missing data. In *Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 461–466). Bruges, Belgium: ESANN.

Peña-Ayala, A. (2017). *Learning Analytics: Fundaments, Applications, and Trends: A View of the Current State of the Art to Enhance e-Learning*. Switzerland: Springer International Publishing.

Regan, K., Berkeley, S., Hughes, M., & Kirby, S. (2014). Effects of computer-assisted instruction for struggling elementary readers with disabilities. *The Journal of Special Education*, 48(2):106–119.

Richardson, U. & Lyytinen, H. (2014). The Graphogame method: The theoretical and methodological background of the technology-enhanced learning environment for learning to read. *Human Technology*,10:39–60.

Rutter, M., Caspi, A., Fergusson, D., Horwood, L., Goodman, R., Maughan, B., Moffitt, T., Meltzer, H., & Carroll, J. (2004). Sex differences in developmental reading disability. *Journal of the American Medical Association*, 291:2007–2012.

Saarela, M., & Kärkkäinen, T. (2015). Weighted clustering of sparse educational data. In *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 337–342). Bruges, Belgium: ESANN.

Saarela, M., & Kärkkäinen, T. (2017). Knowledge discovery from the programme for international student assessment. In A. Peña-Ayala (Ed.), *Learning analytics: Fundaments, applications, and trends: A view of the current state of the art to enhance e-Learning. Studies in systems, decision and control, 94* (pp. 229–267). Switzerland: Springer International Publishing.

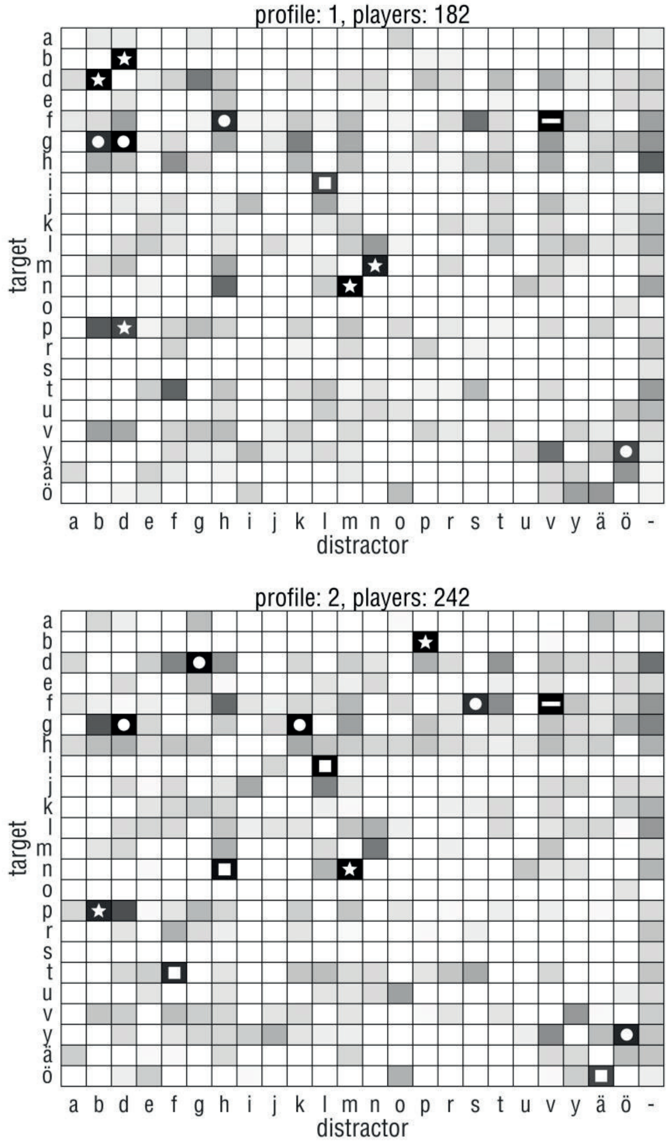
Saine, N., Lerkkanen, M.-K., Ahonen, T., Tolvanen, A., & Lyytinen, H. (2011). Computer-assisted remedial reading intervention for school beginners at risk for reading disability. *Child development*, 82:1013–1028.

Steinbach, M., Ertöz, L., & Kumar, V. (2004). The challenges of clustering high dimensional data. In L. T. Wille (Ed.), *New Directions in Statistical Physics,* (pp. 273–309). Berlin, Heidelberg: Springer.

Wendel, V., Göbel, S., & Steinmetz, R. (2012). Game mastering in collaborative multiplayer serious games. In S. Göbel, W. Müller, B. Urban, & J. Wiemeyer (Eds.), *E-Learning and Games for Training, Education, Health and Sports* (pp. 23–34). Berlin, Heidelberg: Springer.

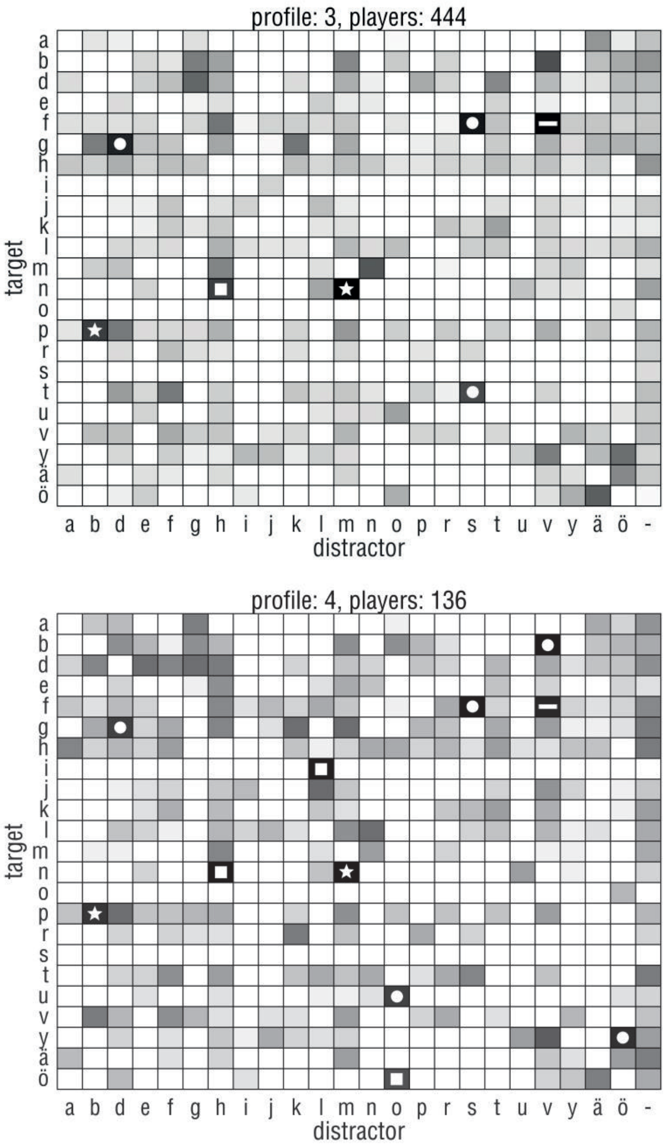
Yanjin, L. & Vincent, A. (2017). Educational game and intelligent tutoring system: A classroom study and comparative design analysis. *ACM Transactions on Computer-Human Interaction,* 24(3):1-27.

Supplement S1: Confusion matrices for lowercase letter data



Darker colors indicate more confusion. Confusions exceeding 10 % are marked with symbols. There are two main categories of letters' similarity: phonetic similarity (marked with a "circle") and visual similarity (marked with a "square"). It is also possible that both or neither categories are occurring (marked with a "star" and a "rectangle", respectively).

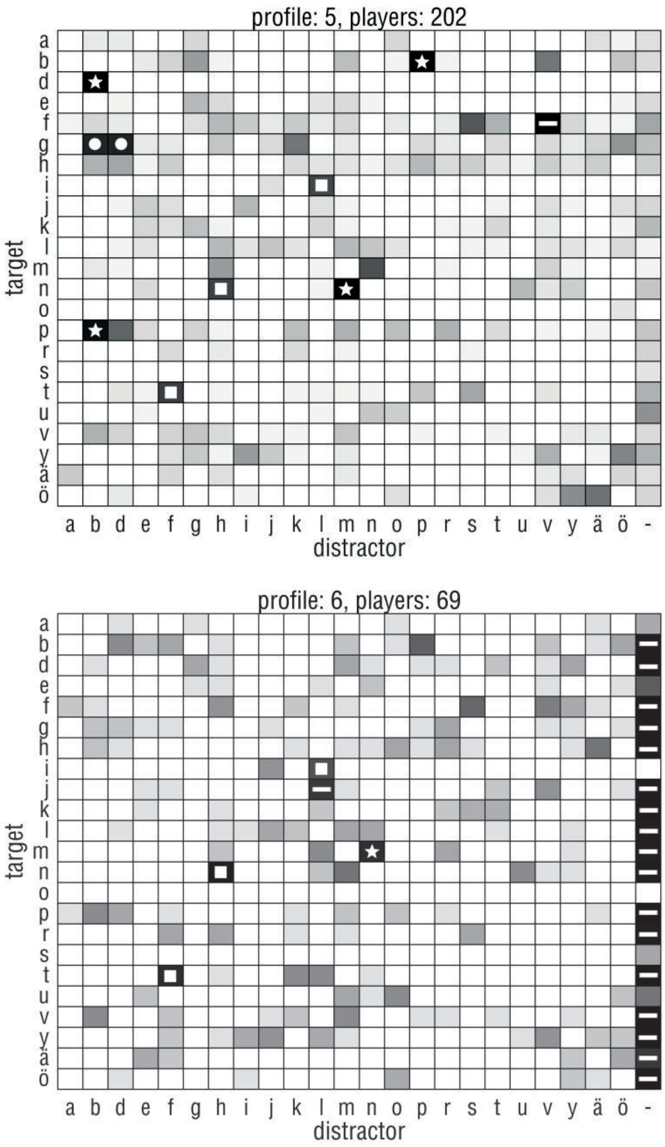
Figure S1.1: Profiles 1 and 2 for lowercase letter data


profile: 3, players: 444

profile: 4, players: 136

Darker colors indicate more confusion. Confusions exceeding 10 % are marked with symbols. There are two main categories of letters' similarity: phonetic similarity (marked with a "circle") and visual similarity (marked with a "square"). It is also possible that both or neither categories are occurring (marked with a "star" and a "rectangle", respectively).

Figure S1.2: Profiles 3 and 4 for lowercase letter data



profile: 5, players: 202

profile: 6, players: 69

Darker colors indicate more confusion. Confusions exceeding 10 % are marked with symbols. There are two main categories of letters' similarity: phonetic similarity (marked with a "circle") and visual similarity (marked with a "square"). It is also possible that both or neither categories are occurring (marked with a "star" and a "rectangle", respectively).

Figure S1.3: Profiles 5 and 6 for lowercase letter data

Supplement S2: K-means clustering and validation indices

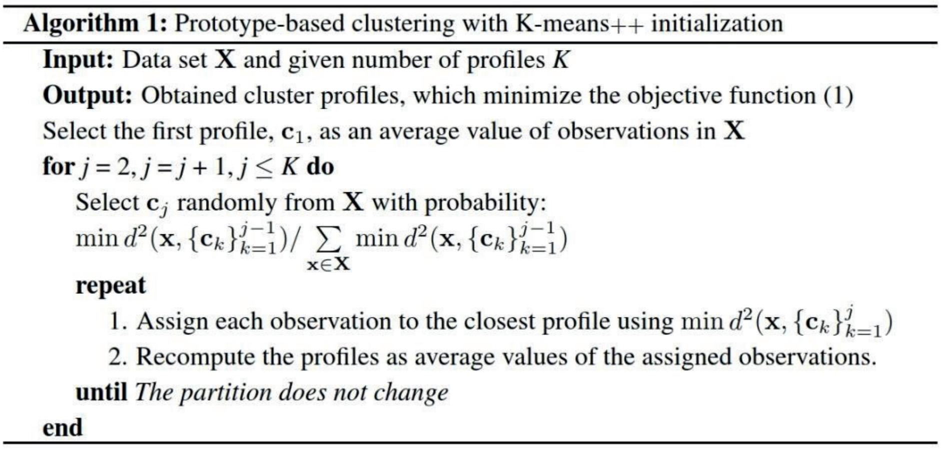
K-means clustering with missing data

The objective function for K-means clustering can be defined as:

$$\arg\min_{\mathbf{C}} \sum_{\mathbf{x}\in\mathbf{X}} d^2(\mathbf{x}, \mathbf{c}_k), \tag{1}$$

where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^n$, refers to a set of N observations, and $\mathbf{C} = \{\mathbf{c}_k\}_{k=1}^{K}$ are obtained cluster profiles. $d()$ denotes modified version of the l_2-norm. Since partially incomplete data, the modified norm is needed for clustering. The main idea of the modified approach is to use pairwise available components and scale the result to the missing components (Gower, 1971) .

K-means clustering method consists of two main steps: an initialization and local refinement steps (see Algorithm 1). These steps are usually performed using multiple restarts and the result with the smallest clustering error will be selected. In an initialization step a local partition of data is decided. The quality of clustering depends on the initialization step since clustering acts locally. A local refinement step perform local search which improve quality of initial partition. The aim of this step is to minimize clustering error, that is, summed distance of observations to the nearest prototypes. The step is performed in an iterative way assigning observations to the nearest prototypes and updating prototype locations. An advantage of K-means with K-means++ type of initializations is that it has only a linear time complexity and comparable fast convergence since K-means++ favors distinct prototypes in a data space (Arthur and Vassilvitskii, 2007).



Algorithm 1: Prototype-based clustering with K-means++ initialization

```
Input: Data set $\mathbf{X}$ and given number of profiles $K$
Output: Obtained cluster profiles, which minimize the objective function (1)
Select the first profile, $\mathbf{c}_1$, as an average value of observations in $\mathbf{X}$
for $j = 2, j = j + 1, j \leq K$ do
    Select $\mathbf{c}_j$ randomly from $\mathbf{X}$ with probability:
    $\min d^2(\mathbf{x}, \{\mathbf{c}_k\}_{k=1}^{j-1}) / \sum_{\mathbf{x}\in\mathbf{X}} \min d^2(\mathbf{x}, \{\mathbf{c}_k\}_{k=1}^{j-1})$
    repeat
        1. Assign each observation to the closest profile using $\min d^2(\mathbf{x}, \{\mathbf{c}_k\}_{k=1}^{j})$
        2. Recompute the profiles as average values of the assigned observations.
    until The partition does not change
end
```

Internal cluster validation indices

In K-means setting the number of clusters is essential to be determined. Internal cluster validation indices (CVIs) identify the number of clusters such that any external/prior information is not needed in the calculations. The most of the CVIs are defined by compactness and separability of the clustering result. The validity index provides a measure for each number of clusters. Depending on the used index formula, the lowest or the highest measure is usually selected as the final number of clusters. Further, the number of clusters can be also selected using the speed of improvement of the cluster validation measures, for example, using a classical knee-point method (Thorndike, 1953).

Our previous study (Niemelä et al., 2018) presented the most commonly used validation indices. The reduced formulas were used since constant terms and monotone functions offered in the original formulas do not affect to the final solutions. In addition, the used formulas were extended for the general similarity measure. In the study, compactness was defined by Intra and separability by Inter. Compactness is usually defined by using summed variances of observations around prototypes in different clusters. Separability indicates how well distinct clusters are for each other. Minimum or maximum values of distances of all prototypes or variance of prototypes are popularly used variables. The study proposed formulas in the form where Intra was divided by Inter and thus they were attempted to be minimized.

In general, the decision of the number of clusters by using CVIs involves the following procedure:

1) Repeat clustering iteratively ranging K from K_{min} to K_{max}. Obtain calculated cluster profiles and data partitions for each value of K based on Algorithm 1.
2) Calculate index measures using CVIs for each value of K. Form index curves based on the measured values.
3) Select the optimal number of clusters according to some decision criteria, for example, minimum/maximum values of cluster validation index curves or using speed of improvements of index measures.

Regarding to the described methods, the source codes are available online: http://users.jyu.fi/~mapeniem/BJET/Kmeans/

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* (pp. 1027–1035). New Orleans, Louisiana, United States: Society for Industrial and Applied Mathematics.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics* (pp. 857-871). Washington, United States: International Biometric Society.

Niemelä, M., Äyrämö, S., & Kärkkäinen, T. (2018). Comparison of cluster validation indices with missing data. In *Proceedings of the 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (pp. 461–466). Bruges, Belgium: ESANN.

Thorndike, R. L. (1953). Who belongs in the family. *Psychometrika, 18*(4): 267–276.

Supplement S3: Results for uppercase letter data

Uppercase data

The analyses of *GraphoLearn* game play data, which was originally performed for the lowercase data set were repeated by using uppercase letter data set. These results are given in Tables S3.1 – S3.2 and Figures S3.1 – S3.2, which can be shortly summarized.

Table S3.1 shows symbols for confusions exceeding 10 % and confusions exceeding 15 % are illustrated with underlined symbols. The most frequently mixed letters were *G*, *D*, *N*, and *M* similarly to the players who used lowercase letter data. Table S3.2 shows error rates from four profiles which were in the range of 34.5 % – 42.7 %. The results are mostly better than the calculated error rates from six profiles of lowercase letter data (30.7 % – 55.3 %). This may be related to fact that uppercase letters are visually less similar than lowercase letters. The progression information was calculated based on only few players because many of players played less than one hour and did not complete the second assessment. Therefore, these numbers give only limited information about the players' progression. The players of this data set have not actively played the game because the total playing times were remarkably smaller and the interval times were higher compared to the times gained from the players who used the lowercase letter data set.

Table S3.1: Symbol table of similarities for different uppercase data profiles

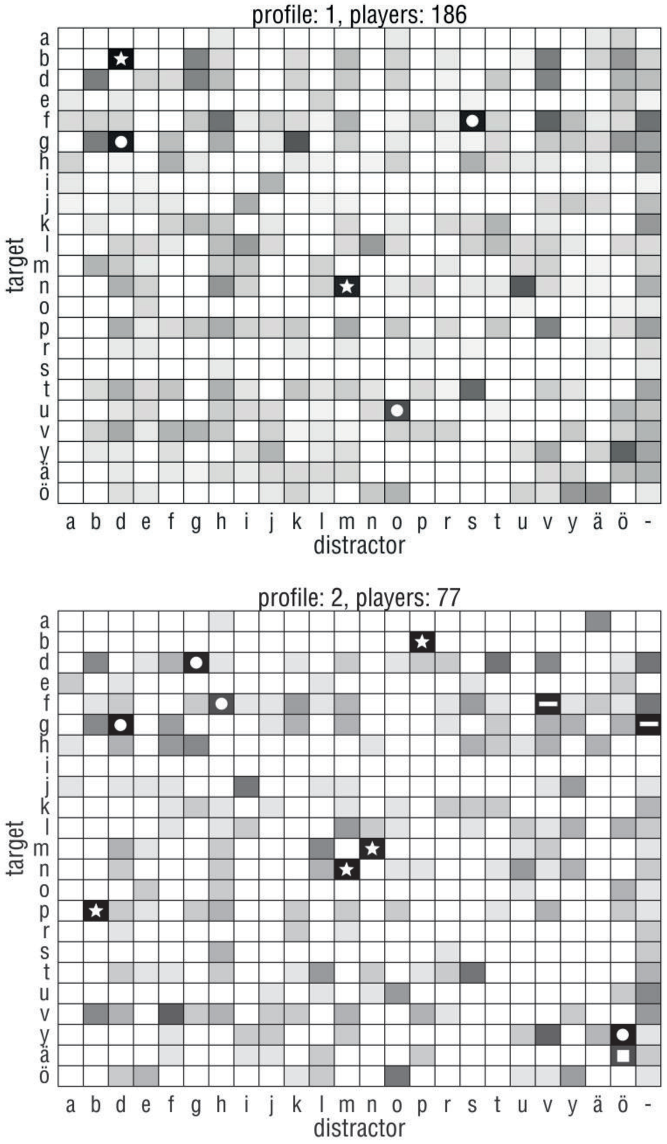
	target | distractor																	
	B|D	B|P	D|B	D|G	F|H	F|S	F|V	G|B	G|D	K|F	M|N	N|M	P|B	U|O	V|F	Y|Ö	Ä|H	Ä|Ö
profile																		
P1	☆					○			○			☆		○				
P2		☆̲		○	○		▭		○̲		☆	☆̲	☆̲			○		□
P3			☆	○				○	○		☆̲	☆̲	☆̲					
P4	☆				○	○̲			○̲	▭	☆̲	☆̲			▭	○	▭	
total	2	1	1	2	2	2	1	1	4	1	3	4	2	1	1	2	1	1

phonetic similarity=○, visual similarity=□, phonetic and visual similarity=☆,
unknown category=▭

Table S3.2: Findings of uppercase data profiles

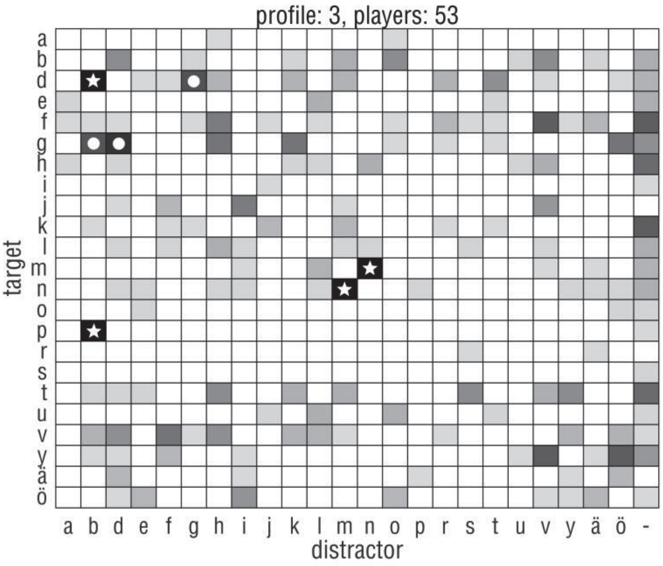
	profile				**all**
	P1	P2	P3	P4	P
statistics					
size (in %)	52.8%	21.9%	15.1%	10.2%	100.0%
error rate	34.5%	38.7%	36.2%	42.7%	36.5%
progression*	10.2%	23.0%	13.1%	11.9%	13.8%
playing time	36.8 min	46.5 min	38.2 min	29.7 min	39.5 min
interval time	7.0 days	8.1 days	7.0 days	9.5 days	7.5 days

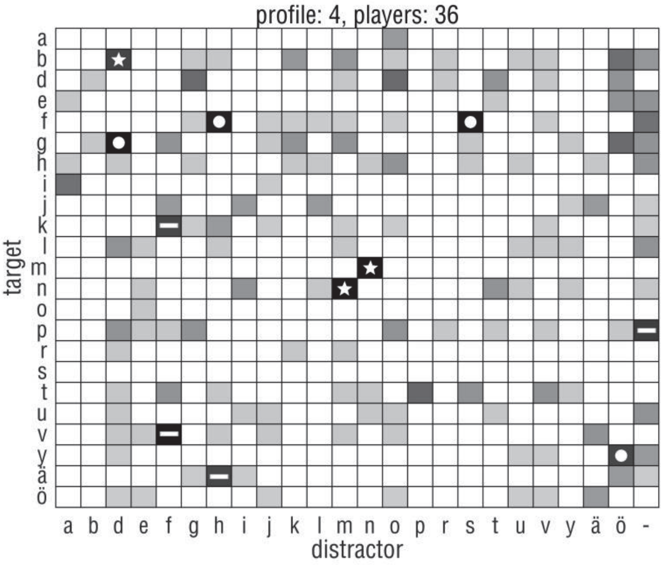
*Only players who completed both assessments are included.



Darker colors indicate more confusion. Confusions exceeding 10 % are marked with symbols. There are two main categories of letters' similarity: phonetic similarity (marked with a "circle") and visual similarity (marked with a "square"). It is also possible that both or neither categories are occurring (marked with a "star" and a "rectangle", respectively).

Figure S3.1: Profiles 1 and 2 for uppercase letter data





Darker colors indicate more confusion. Confusions exceeding 10 % are marked with symbols. There are two main categories of letters' similarity: phonetic similarity (marked with a "circle") and visual similarity (marked with a "square"). It is also possible that both or neither categories are occurring (marked with a "star" and a "rectangle", respectively).

Figure S3.2: Profiles 3 and 4 for uppercase letter data

II

COMPARISON OF CLUSTER VALIDATION INDICES WITH MISSING DATA

by

Marko Niemelä, Sami Äyrämö, and Tommi Kärkkäinen 2018

ESANN 2018, proceedings on European Symposium on Artificial Neural
Networks, Computational Intelligence, and Machine Learning, pages 461–466

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Comparison of Cluster Validation Indices with Missing Data

Marko Niemelä[1,2]*, Sami Äyrämö[1] and Tommi Kärkkäinen[1]†

1- Faculty of Information Technology, University of Jyvaskylä
PO Box 35, FI-40014 Jyväskylä, Finland

2- Niilo Mäki Institute
PO Box 35, FI-40014 Jyväskylä, Finland

Abstract.
Clustering is an unsupervised machine learning technique, which aims to divide a given set of data into subsets. The number of hidden groups in cluster analysis is not always obvious and, for this purpose, various cluster validation indices have been suggested. Recently some studies reviewing validation indices have been provided, but any experiments against missing data are not yet available. In this paper, performance of ten well-known indices on ten synthetic data sets with various ratios of missing values is measured using squared euclidean and city block distances based clustering. The original indices are modified for a city block distance in a novel way. Experiments illustrate the different degree of stability for the indices with respect to the missing data.

1 Introduction

In clustering, a given set of data is divided into subsets, clusters, such that observations in a cluster are similar to each other and dissimilar to observations in the other clusters. Even though the principle is simple, there exist multiple clustering approaches [1] of which the main groups are prototype-based and hierarchical clustering. Prototype-based algorithms, such as K-means [2], utilize error functions based on within-cluster distances, which then provide data partition with location estimates, e.g., the sample mean, as the cluster prototypes. K-medians is a robust variant of K-means algorithm, which does not assume spherically symmetric, normally distributed cluster shapes, but instead the variables can consist of discrete values with uniform quantization error [3]. Further, another property of K-medians is robustness against outliers since the breakdown point of the median is 50 %.

Prototype-based clustering typically requires the number of clusters, denoted by K, as an input parameter. Determining the correct number of clusters is a difficult task, because there are often more than one possible solutions to a clustering problem. The existing methods to estimate the number of clusters are based on, e.g., visual evaluation of clustering error [4], stability of the solution

*The work has been supported by the Academy of Finland from the project 311737 (DysGeBra)

†The work has been supported by the Academy of Finland from the projects 311877 (Demo) and 315550 (HNP-AI)

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence
and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

[5], and multiobjective evolutionary algorithms [6]. Cluster validation indices analyze the quality of clustering models by assessing compactness and separability of clusters with different values of K.

Internal cluster validation indices have been compared in recent studies. In [7], `kCE-index` was found to be the best performing index over 43 indices, being the only index able to validate successfully the single cluster data set, in which the other indices recommended higher numbers. In [8], `Wemmert-Gançarski` outperformed other indices when three distance measures and clustering approaches with 56 synthetic and 6 real world data sets were used. The study summarized different results for different indices. For some indices, the performances varied between different distances. In [9], `Silhouette` index was generally the best of 30 indices through a large number of experiments, including demanding data sets with high dimensionalities, noise, and overlapping clusters.

Despite the extensive comparisons of indices in the previous studies, none of them considered data sets with missing values. However, missing values are common in the real-world data. There could be a variety of reasons to explain missingness of variables, including measurement error, device malfunction, unanswered question, etc. Many clustering approaches are based on the assumption of complete data sets, therefore, such methods cannot be applied directly if some of the data values are missing.

In this work, the previous work especially in [7, 8] was continued by selecting the best performing indices to the comparison. The original indices based on euclidean distance were extended also for city block distance. The selected clustering methods and indices, presented in Section 2, were developed to be tolerant for missing values. Numerical results demonstrating the quality of indices are given and the main findings are discussed in Section 3.

2 Methods

The prototype-based clustering methods consist of an initialization step, in which an initial partition of the data is decided, and a local refinement step, in which the quality of the initial partition is improved by an iterative local search algorithm. Hence, in a general case, the following clustering error is minimized during the local search:

$$\mathcal{J}(\{\mathbf{c}_k\}) = \sum_{i=1}^{N} \min_{k=1,\dots,K} \|\mathbf{x}_i - \mathbf{c}_k\|_p^q = \sum_{k=1}^{K} \mathcal{J}_{p,k}^q = \mathcal{J}_p^q, \tag{1}$$

where $\{\mathbf{x}_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^n$, is the given set of n-dimensional observations, N is the number of observations, and $\{\mathbf{c}_k\}_{k=1}^K$ are the obtained prototype vectors. l_p-norms to q-th power are utilized for different location estimates. The within-cluster error in cluster C_k, is denoted by $\mathcal{J}_{p,k}^q$ and the total residual error of a local minimizer of Eq. 1 is denoted by $\mathcal{J}_p^q$. By choosing $p = q = 1$ or $p = q = 2$, the error function for K-medians or K-means, respectively, are obtained. Note that if $q = 1$ it can be omitted from the notation.

In this study, a partial distance strategy for calculating distances is adopted from [10] since the data vectors may consist missing values. The idea is that the sum of differences of the known components are used and scaled to the missing components. The original method was developed for the l_2-norm, but a modified version for the l_1-norm is offered in the current study. Distances based on l_1 and l_2 norms read as $\hat{d_1}(\mathbf{x}, \mathbf{y}) = \frac{n}{\hat{n}} \sum_{j=1}^{\hat{n}} |(\mathbf{x})_j - (\mathbf{y})_j|$ and $\hat{d_2}(\mathbf{x}, \mathbf{y}) = \sqrt{\frac{n}{\hat{n}} \sum_{j=1}^{\hat{n}} ((\mathbf{x})_j - (\mathbf{y})_j)^2}$, respectively. $\hat{n}$ indicates the number of components that exist in both of the compared vectors. We assume that $\hat{n} > 0$. The modified version of Eq. 1 is required due to missing data. The new estimated clustering error, based on the partial distance strategy, is defined as $\hat{\mathcal{J}}_p^q = \sum_{i=1}^{N} \min_{k=1,\dots,K} \hat{d}_p^q(\mathbf{x}_i, \mathbf{c}_k)$.

Internal cluster validation indices prefer both high within clusters similarity and between clusters separability. In this work, the measured within-cluster similarity is referred to as *Intra* and between-cluster separability as *Inter*. Low values are better for *Intra* and high values for *Inter*. The optimal solution is obtained by minimizing or maximizing the ratio of Intra and Inter measures.

The eight best performing incides from [7] in addition to `WB-index (WB)` [8] and `Davies-Bouldin`* [9] were compared in this study. All the indices, except `Silhouette`, are defined in Table 1. We presented general forms of reduced formulas, where constant terms or monotone functions have been omitted. The formulas are attempted to be minimized since *Intra* is divided by *Inter*. The clustering error is often used as *Intra*. Further, many indices tend to define *Inter* as the minimum distance between cluster prototypes. Distances between cluster prototypes and the whole data prototype are also commonly applied as *Inter* value. In addition, `WB`, `Calinski-Harabasz`, and `kCE-index` utilize penalization terms for a high number of clusters that were originally defined in the context of the squared euclidean distance. Initial experiments showed that these terms penalized too much while non-squared counterparts were used, therefore, square roots over terms were taken in these cases.

In `Silhouette` index, *Intra* is the average dissimilarity of $\mathbf{x}_i$ to all other points in the same cluster and *Inter* is the minimum average dissimilarity of $\mathbf{x}_i$ to all points in a different cluster. Silhouette index is defined as $\sum_{i=1}^{N} \frac{Inter(\mathbf{x}_i) - Intra(\mathbf{x}_i)}{\max(Intra(\mathbf{x}_i), Inter(\mathbf{x}_i))}$. Contrary to indices that use full prototypes for calculating an index value with missing data, `Silhouette` calculates distances between observations that are sometimes incomplete. Hence, the adopted distance calculation technique, presented in this study, is especially beneficial for `Silhouette` since there is always a higher risk that at least one of pairwise components is missing.

Ten synthetic data sets were used in the study. Four *S*[1] sets and two *D*[1] sets were selected from [11]. *Sim2D2*[2] and *Sim5D2*[2] data sets were selected from [7]. New similar *O200*[2] and *O2000*[2] data sets with a different number of observations were created for this study. Both *O* data sets consist of five clusters in total, one Gaussian and four Laplace distributed clusters. In addition, 10 % of uniformly

[1] http://cs.uef.fi/sipu/datasets/

[2] http://users.jyu.fi/˜mapeniem/CVI/Data/

distributed noise was added to new data sets. D sets are 32 and 256 dimensional and the other presented data sets are two dimensional.

Table 1: Formulas of cluster validation indices.

Name	**Intra**	**Inter**	**Formula**
Calinski-Harabasz (CH)	$\hat{\mathcal{J}}_p^p$	$\sum_{k=1}^{K} n_k \lVert \mathbf{c}_k - \mathbf{m} \rVert_p^p$	$(\frac{K-1}{N-K})^{\frac{1}{3-p}} \times \frac{\mathit{Intra}}{\mathit{Inter}}$
Davies-Bouldin (DB)	$\frac{\hat{\mathcal{J}}_{p,k}}{n_k} + \frac{\hat{\mathcal{J}}_{p,k'}}{n_{k'}}$	$\lVert \mathbf{c}_k - \mathbf{c}_{k^*} \rVert_p$	$\frac{1}{K} \sum_{k=1}^{K} \max_{k \neq k'} \frac{\mathit{Intra}(k,k')}{\mathit{Inter}(k,k')}$
Davies-Bouldin* (DB*)	$\frac{\hat{\mathcal{J}}_{p,k}}{n_k} + \frac{\hat{\mathcal{J}}_{p,k'}}{n_{k'}}$	$\lVert \mathbf{c}_k - \mathbf{c}_{k^*} \rVert_p$	$\frac{1}{K} \sum_{k=1}^{K} \frac{\max_{k \neq k'} \mathit{Intra}(k,k')}{\min_{k \neq k^*} \mathit{Inter}(k,k^*)}$
Generalized Dunn (GD)	$\max \frac{\hat{\mathcal{J}}_{p,k}}{n_k}$	$\min_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p$	$\frac{\mathit{Intra}}{\mathit{Inter}}$
kCE-index (KCE)	$\hat{\mathcal{J}}_p^p$	1	$K^{\frac{1}{3-p}} \times \mathit{Intra}$
Pakhira-Bandyopadhyay-Maulik (PBM)	$\hat{\mathcal{J}}_p$	$\max_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p,$	$K \times \frac{\mathit{Intra}}{\mathit{Inter}}$
Ray-Turi (RT)	$\hat{\mathcal{J}}_p^p$	$\min_{k \neq k'} \lVert \mathbf{c}_k - \mathbf{c}_{k'} \rVert_p^p$	$\frac{\mathit{Intra}}{\mathit{Inter}}$
WB-index (WB)	$\hat{\mathcal{J}}_p^p$	$\sum_{k=1}^{K} n_k \lVert \mathbf{c}_k - \mathbf{m} \rVert_p^p$	$K^{\frac{1}{3-p}} \times \frac{\mathit{Intra}}{\mathit{Inter}}$
Wemmert-Gançarski (WG)	$\hat{d}_p(\mathbf{x}_i, \mathbf{c}_k)$	$\min_{k \neq k'} \hat{d}_p(\mathbf{x}_i, \mathbf{c}_{k'})$	$\sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \frac{\mathit{Intra}(\mathbf{x}_i)}{\mathit{Inter}(\mathbf{x}_i)}$

3 Experimental results and conclusion

Experiments were performed using MATLAB (R2015b, 64-bit). Data sets were min-max scaled to a range of [-1, 1] before clustering and index value calculations. Incomplete data sets with varying numbers of missing values were created by removing data values completely at random from the existing test data sets. The clustering was repeated 100 times from random initial conditions of prototypes and the solution of the lowest local minima was selected as the final solution. The initialization was performed in an iterative manner such that K ranged from 2 to 20. More specifically, the obtained prototypes were saved for each K and these previously saved prototypes were utilized during the next initialization. The generalized version of K-means++ algorithm (see [8] for details) was used and therefore the next prototype was selected based on the calculated distances to the closest already selected prototypes such that the most distant point had the highest probability of being selected.

Table 2 shows the obtained results. Clearly, `WG` and `Silhouette` were generally the two best performing indices suggesting 64 and 63 correct solutions in total, respectively. Further, `WG`, `KCE`, and `CH` were the three best performing indices for the euclidean distance, giving 36, 33, and 33 correct solutions, respectively. In addition, `Silhouette` and `WG` were the two best ones for the city block distance, proposing 31 and 28 correct solutions, respectively. Regarding the stability of indices, `WG` showed to be the most stable, giving always nine correct solutions over ten data sets for the euclidean distance while the proportion of missing data was gradually increased from 0 % to 20 %. `CH` was the stable index for the city block. However, it only offered six correct solutions for each level of missing values. For the most of indices, especially for euclidean distance based indices, the high number of missing values has negative impact on the performance. As shown in Table 2, the whole clustering algorithm did not cause instability to the index results since only in four cases the correct number of clusters was not found after clustering with random initial prototypes, but only after using the known centers, given by the authors of the data sets, as initial prototypes in clustering.

This section provided results which were obtained when cluster validation indices were compared. Previous studies [7, 8] were continued by extending clustering methods and indices to city block distances and to handle missing values. Similarly to the previous studies, `WG`, `Silhouette`, and `KCE` were nominated to be the best performing indices in this study. All indices performed better with the euclidean distance compared to the city block distance. The used data sets are all continuous valued which may explain the better results with the euclidean distance. `Silhouette` produced almost identical results for these two distances and was the best index for the city block. Different stability patterns for the indices were shown in the study. `WG` was the most stable index, recommending nearly always the same numbers for clusters over the different levels of missing values. Future research direction is to use real-world data in experiments. Further testing is also needed with multidimensional data since all the indices offered always correct answers for *D32* and *D256* data sets.

References

[1] C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications.* CRC press, 2013.

[2] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[3] M. Saarela and T. Kärkkäinen. Analysing student performance using sparse data of core bachelor courses. *JEDM-Journal of Educational Data Mining*, 7(1):3–32, 2015.

[4] R. L. Thorndike. Who belongs in the family. *Psychometrika*, pages 267–276, 1953.

[5] L. I. Kuncheva and D. P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1798–1808, 2006.

[6] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. P. L. F. de Carvalho. A survey of evolutionary algorithms for clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):133–155, 2009.

[7] S. Jauhiainen and T. Kärkkäinen. A simple cluster validation index with maximal coverage. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning - ESANN 2017*, pages 293–298, 2017.

[8] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen. Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 10(3), 2017.

[9] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, 2013.

[10] J. Gower. A general coefficient of similarity and some of its properties. 27:857–871, 1971.

[11] Q. Zhao and P. Fränti. Wb-index: A sum-of-squares based index for cluster validity. *Data & Knowledge Engineering*, 92:77–89, 2014.

Euc **Cit**	***CH***	***DB***	***DB****	***GD***	***KCE***
S1	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
	15 15 15 15	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
S2	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
	15 15 15 15	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
S3	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	4 4 **15** 4	**15 15 15 15**
	15 15 15 15	7 14 14 14	4 4 4 4	4 4 4 4	**15 15 15** 16
S4	**15 15 15 15**	14 14 14 17	13 13 13 13	4 4 4 4	**15 15 15 15**
	15 15 15 15	17 17 17 17	4 4 4 4	4 4 4 4	**15 15 15** 16
D32	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
	16 16 16 16	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
D256	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
	16 16 16 16	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
Sim2D2	**2 2 2 2**	**2 2** 20 19	**2 2 2 2**	**2 2 2 2**	**2 2 2 2**
	4 4 4 20	13 13 20 18	**2 2 2 2**	**2 2 2 2**	**2 2 2** 20
Sim5D2	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	**5 5 5 5**
	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3	3 3 3 3
O200	**5** 20 20 20	**5 5 5** 20	**5 5 5 5**	4 4 4 4	20 20 20 20
	20 20 20 20	8 8 20 20	8 **5** 4 **5**	**5 5 5 5**	20 20 20 20
O2000	**5 5 5 5**	**5 5 5 5**	**5 5 5 5**	**5 5 5 5**	**5** 6 6 7
	6 12 13 20	6 6 6 **5**	4 4 4 **5**	4 4 4 **5**	1 6 6 14
Total	**9 8 8 8**	**8 8 7 6**	**8 8 8 8**	**6 6 7 6**	**9 8 8 8**
	6 6 6 6	**4 4 4 5**	**5 6 5 7**	**6 6 6 7**	**7 7 7 4**
	PBM	***RT***	***SIL***	***WB***	***WG***
S1	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
	15 15 15 15	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
S2	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
	15 15 15 15	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**	**15 15 15 15**
S3	5 5 4 4	4 4 4 4	**15 15** 2 2	**15 15 15** 16	**15 15 15 15**
	4 4 4 4	4 4 **15 15**	**15 15 15** 2	**15 15 15** 16	**15 15 15 15**
S4	4 4 4 4	13 13 10 10	**15 15 15** 3	**15 15 15** 20	**15 15 15 15**
	5 5 5 5	17 17 4 14	**15 15** 14 14	**15 15 15** 16	16 16 16 16
D32	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
	16 16 16 16	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
D256	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
	16 16 16 16	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**	**16 16 16 16**
Sim2D2	**2 2 2 2**	**2 2 2 2**	**2 2 2 2**	12 12 20 20	**2 2 2 2**
	2 2 2 2	**2 2 2 2**	**2 2 2 2**	4 20 20 20	**2 2 2 2**
Sim5D2	**5 5 5 5**	3 3 3 3	3 3 3 3	**5 5 5 5**	3 3 3 3
	5 5 4^+ 4^+	3 3 3 3	3 3 3 3	4 7 7 17	3 3 3 3
O200	**5 5 5 5**	**5 5 5 5**	**5 5 5 5**	20 20 20 20	**5 5 5 5**
	3 3 3 3	**5 5 5 5**	**5 5 5 5**	20 20 20 20	**5 5** 20 20
O2000	**5** 4 4 4	**5 5 5** 4	**5 5 5** 6	6 7 7 20	**5 5 5 5**
	3 3 3 3	4 4 4 **5**	**5 5** 6^+ 6^+	14 13 20 20	**5 5** 6 2
Total	**8 7 7 7**	**7 7 7 6**	**9 9 8 6**	**7 7 7 5**	**9 9 9 9**
	6 6 5 5	**6 6 7 8**	**9 9 7 6**	**6 6 6 4**	**8 8 6 6**

$^+$ Result can be corrected using the known centers as initial prototypes

Table 2: The determined number of clusters by cluster validation indices. The correct numbers are bolded. The results are given in four columns, one column for each percentage (0, 5, 10, and 20 %) of missing values.

III

IMPROVING CLUSTERING AND CLUSTER VALIDATION WITH MISSING DATA USING DISTANCE ESTIMATION METHODS

by

Marko Niemelä and Tommi Kärkkäinen 2022

Computational Sciences and Artificial Intelligence in Industry: New Digital Technologies for Solving Future Societal and Economical Challenges, pages 123–133

Reproduced with kind permission of Springer International Publishing.

https://doi.org/10.1007/978-3-030-70787-3_9

Improving Clustering and Cluster Validation with Missing Data using Distance Estimation Methods

Marko Niemelä and Tommi Kärkkäinen

Abstract Missing data introduces a challenge in the field of unsupervised learning. In clustering, when the form and the number of clusters is to be determined, one needs to deal with the missing values both in the clustering process and in the cluster validation. In the previous research, the clustering algorithm has been treated using robust clustering methods and available data strategy, and the cluster validation indices have been computed with the partial distance approximation. However, lately special methods for distance estimation with missing values have been proposed and this work is the first one where these methods are systematically applied and tested in clustering and cluster validation. More precisely, we propose, implement, and analyze the use of distance estimation methods to improve the discrimination power of clustering and cluster validation indices. A novel, robust prototype-based clustering process in two stages is suggested. Our results and conclusions confirm the usefulness of the distance estimation methods in clustering but, surprisingly, not in cluster validation.

1 Introduction

The two main approaches for prototype-based clustering with missing values are imputation (Lin and Tsai [11]) and available data strategy. Combined with a statistically robust (see Kärkkäinen and Heikkola [9]) cluster prototypes like median or spatial median (Äyrämö [2]), the available data strategy

Marko Niemelä
University of Jyväskylä, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyväskylä, Finland, e-mail: marko.p.niemela@jyu.fi

Tommi Kärkkäinen
University of Jyväskylä, Faculty of Information Technology, P.O. Box 35, FI-40014 University of Jyväskylä, Finland e-mail: tommi.karkkainen@jyu.fi

has proven to provide reliable results in a scalable fashion (Hämäläinen et al. [8]). However, in many applications the unsupervised tasks that need to be solved consist of estimation and determination of both the clusters and the number of them. The latter is addressed using cluster validation indices, which have been scarcely addressed with missing values although new techniques constantly emerge (Fu and Perry [5]).

As depicted in Hämäläinen et al. [7], Niemelä et al. [13], the cluster validation indices are composed of a quotient of estimates of *Inter* and *Intra* of a clustering result, i.e., the variability of data within clusters divided by the separation of clusters. Both of these measures are computed with a distance measure which is inhereted from the clustering problem formulation (Hämäläinen et al. [8]). Therefore, a key to reliable cluster validation indices with missing values is how to estimate the distances between the prototypes and the observations. For this purpose, in Niemelä et al. [13], the classical partial distance strategy (Gower [6]) was applied with promising results. However, more recently a set of papers have appeared (Eirola et al. [3, 4], Mesquita et al. [12]), which have addressed the distance estimation with missing values for both squared and euclidean (nonsquared) distances with better accuracy than in Gower [6].

This work continues the work in Niemelä et al. [13] by offering similar comparisons of cluster validation indices when the clustering method is replaced with the use of l_2-norm, i.e., optimized values of cluster prototypes minimize the Euclidean distance error with the target data instead using the squared Euclidean distance based error function (Äyrämö [2], Hämäläinen et al. [7]). Further, instead of the partial distance strategy, we utilize two previously presented distance estimation strategies (Eirola et al. [3], Mesquita et al. [12]) for calculating the distances between the possible incomplete data vectors during the cluster evaluation process. A novel, robust prototype-based clustering process in two stages is suggested when these strategies are applied in clustering. We then assess the usefulness of the distance estimation in cluster validation. As a whole, the purpose of this paper is to realize and test the distance estimation methods in an attempt to improve the reliability of clustering and cluster validation indices with missing values.

2 Methods

Prototype-based clustering methods, such as K-means, solve an optimization problem with K prototypes (Äyrämö [2], Hämäläinen et al. [7]). The objective function is defined to minimize the sum of the distances of the points to their closest prototypes. The prototype-based algorithm is composed of initialization and local improvement of the initial prototypes. This refinement is carried out in an iterative fashion by assigning individual observations to the closest prototypes and recomputation of the prototype with the assigned observations. These steps are repeated until the final converge is reached (Hämäläinen et al. [7]). The initial prototypes can be selected randomly but a more effective method is to use the K-means++ type of initial selection (Arthur and Vassilvitskii [1], Hämäläinen et al. [7]).

Spatial median is a statistically robust location estimate which can tolerate a large amount of missing values in data since it can handle up to 50 % of erroneous or missing components (Äyrämö [2]). The available data strategy (ADS) is a convenient way to omit the missing values during the cluster refinement phase. It is based on projecting all computations to the available values using a projection matrix $\mathbf{P}$, which represent the pattern of the available values similarly to Kärkkäinen and Toivanen [10]. This is obtained by setting $(\mathbf{P}_i)_j = 1$ if and only if the corresponding data component $(\mathbf{x}_i)_j$ exists, and zero otherwise. Using the available data strategy, the objective function for the spatial median based clustering can be written as follows:

$$\mathcal{J} = \sum_{k=1}^{K} \mathcal{J}_k = \underset{\{\mathbf{c}_k\}}{\arg\min} \sum_{\mathbf{x}_i \in \mathbf{C}_k} \|\mathbf{P}_i\left(\mathbf{x}_i - \mathbf{c}_k\right)\|\,, \tag{1}$$

where $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^n$, is the set of N observations with n-dimensions and $\{\mathbf{c}_k\}_{k=1}^K$ are the prototype vectors which are local minimizers of (1) defining the partition $\mathbf{C}_{k=1}^K$ of data into K disjoint subsets. We emphasize that the base of ADS, realized through the projection, lies in avoiding to introduce any additional assumptions on the data distribution.

In Eirola et al. [3], the expected squared Euclidean distance (ESD) estimation method for missing data was presented. The method assumes multivariate normally distributed data, which may be valid in many real world situations. Normality provides a rough approximation for nearly any continuous data distribution with relevant sample size, e.g., due to the central limit theorem (Rouaud [14]). In particular, it is assumed in Eirola et al. [3] that missing values in data vectors are random variables from the conditional normal distribution in which random variables are conditioned with the observed ones. In this case the incomplete parts of the vectors can be replaced with the conditional mean. If the missing components of $\mathbf{x}$ are denoted by $\mathbf{x}^{(1)}$ and the available components are denoted by $\mathbf{x}^{(2)}$ and n-dimensional incomplete multivariate data is partitioned as follows:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \end{bmatrix}, \tag{2}$$

then

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}^{(1)} \\ \boldsymbol{\mu}^{(2)} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix},$$

where $\boldsymbol{\mu}$ and Σ denotes mean and covariance of $\mathbf{x}$. Further, conditional mean and variance for missing values can be expressed as follows:

$$\begin{aligned}\hat{\mathbf{x}}^{(1)} &= \boldsymbol{\mu}^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}^{(2)} - \boldsymbol{\mu}^{(2)}),\\ (\sigma^2)^{(1)} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.\end{aligned}$$

Notice that the multivariate normal parameters can be estimated for data sets even data cannot pre-partitioned as in (2). Thus, for conditional parameters, appropriate elements are required to be extracted from specific locations in $\boldsymbol{\mu}$ and Σ based on missingness pattern of individual observations.

It was proved in Eirola et al. [3] that the expected value for the squared Euclidean distance is the sum of the distance between the two estimated data vectors and the variances of the imputed components:

$$E[d_{il}^2] = E[||\mathbf{x}_i - \mathbf{x}_l||^2] = ||\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_l||^2 + \sigma_i^2 + \sigma_l^2,\ i \neq l,\ i, l \in [1, N].$$

A novel expected Euclidean distance (EED) method for estimating the nonsquared l_2-norm based distances with missing values was presented in a more recent study Mesquita et al. [12]. It uses the same basic principles as in Eirola et al. [3] for calculating the conditional distribution parameters. However, the EED is based on the assumption that the squared variables follow the Gamma distribution. This suggests use of the Nakagami distribution, where a random variable is obtained by taking the square root of a Gamma distributed variable. More precisely, the expected value of the Nakagami distribution can be written as

$$E[d_{il}] = \frac{\Gamma(m + \frac{1}{2})}{\Gamma(m)} \left(\frac{\Omega}{m}\right)^{\frac{1}{2}}, \tag{3}$$

where

$$m = \frac{E[d_{il}^2]^2}{Var[d_{il}^2]}, \quad \Omega = E[d_{il}^2].$$

Since the Nakagami distribution requires variances of distances, some extra calculations are needed. The variances can be calculated as follows (the details are given in Mesquita et al. [12]):

$$\mathrm{Var}[d_{il}^2] = E[\mathbf{x}_i^4 + \mathbf{x}_l^4 - 4\mathbf{x}_i^3\mathbf{x}_l - 4\mathbf{x}_i\mathbf{x}_l^3 + 6\mathbf{x}_i^2\mathbf{x}_l^2] - E[(\mathbf{x}_i - \mathbf{x}_l)^2]^2,$$

where the expected values can be obtained by using non-central moments of the normal distribution:

$$\begin{aligned}E[\mathbf{x}_i] &= \hat{\mathbf{x}}_i,\\ E[\mathbf{x}_i^2] &= \hat{\mathbf{x}}_i{}^2 + \sigma_i^2,\\ E[\mathbf{x}_i^3] &= \hat{\mathbf{x}}_i{}^3 + 3\hat{\mathbf{x}}_i\sigma_i^2,\\ E[\mathbf{x}_i^4] &= \hat{\mathbf{x}}_i{}^4 + 6\hat{\mathbf{x}}_i{}^2\sigma_i^2 + 3\sigma_i^4.\end{aligned}$$

Notice that we do not apply the weighted formulas in Mesquita et al. [12], because we assume similarly to ESD that the distributions are multivariate Gaussians instead of mixture of Gaussians.

Concerning cluster validation, we will apply the same cluster validation indices as in our previous study Niemelä et al. [13]. References to the original suggestions of the indices are given in Hämäläinen et al. [7], Niemelä et al. [13]. These read as follows (abbreviations given in parenthesis): `Calinski-Harabasz` (`CH`), `Davies-Bouldin` (`DB`), `Davies-Bouldin`* (`DB`*), `Generalized Dunn` (`GD`), `kCE-index` (`KCE`), `Pakhira-Bandyopadhyay-Maulik` (`PBM`), `Ray-Turi` (`RT`), `Silhouette` (`SIL`), `WB-index` (`WB`), and `Wemmert-Gançarski` (`WG`). Since clustering here is performed using the Euclidean distances (1), the indices were first implemented and preliminary tested by using the l_2-norm. We then noticed that `Calinski-Harabasz`, `kCE-index`, and `WB-index` obtained better results with their original forms of using the squared distances in the definitions of *Intra* and *Inter*. The reason might be that these indices include a scaling factor which was originally derived for the squared distances.

The formulas for the used indices are given in Table 1. There, $\mathbf{m}$ denotes the spatial median of the whole dataset. Moreover, the squared form $(\cdot)^2$ can also denote a componentwise application, for instance, within each cluster for $\mathcal{J}_k$ as in (1). We remind that the main focus of this work is that the distances both in clustering and in the CVIs afterwards can be computed with ADS, ESD, or EED, respectively.

In the `Silhouette` index, $\mathit{Intra}(\mathbf{x}_i)$ is the average Euclidean distance of the ith observation to all other points in the same cluster whereas $\mathit{Inter}(\mathbf{x}_i)$ is the average of the minimum distances of the ith point to points in a different cluster:

$$\mathit{Intra}(\mathbf{x}_i) = \frac{1}{n_k - 1} \sum_{\mathbf{x}_j \in \mathbf{C}_k} d(\mathbf{x}_i, \mathbf{x}_j), \;\; \mathit{Inter}(\mathbf{x}_i) = \min_{k \neq k'} \frac{1}{n_{k'}} \sum_{\mathbf{x}_j \in \mathbf{C}_{k'}} d(\mathbf{x}_i, \mathbf{x}_j). \tag{4}$$

Contrary to other indices in Table 1, in `Silhouette` one needs to calculate pairwise distances between the original, possible incomplete observations. Hence, the distance estimation techniques as presented above could be especially beneficial for the `Silhouette` index. On the other hand, because of the computations over each cluster and each observation within a cluster, the computational complexity is of the order $\mathcal{O}(N^2)$.

Notice that in Table 1 both *Intra* and *Inter* can be defined in three levels of abstraction concerning the clustering result: globally as, e.g., with `kCE-index`, clusterwise as, e.g., with `Davies-Bouldin`, and pointwise as, e.g., in `Silhouette`. This division is reflected in the actual Formula where no arguments is being given in the global case (*Intra* in `kCE-index`), arguments related to clusters are given in the clusterwise case ($\mathit{Intra}(k, k')$ in `Davies-Bouldin`), and index of an individual observation is given in the final case ($\mathit{Intra}(\mathbf{x}_i)$ in `Silhouette`), respectively.

Table 1 Formulas of cluster validation indices

Abbr	Intra	Inter	Formula
`CH`	$\mathcal{J}^2$	$\sum_{k=1}^{K} n_k d(\mathbf{c}_k, \mathbf{m})^2$	$\frac{K-1}{N-K} \times \frac{Intra}{Inter}$
`DB`	$\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}}$	$d(\mathbf{c}_k, \mathbf{c}_{k^*})$	$\frac{1}{K} \sum_{k=1}^{K} \max_{k \neq k'} \frac{Intra(k,k')}{Inter(k,k')}$
`DB*`	$\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}}$	$d(\mathbf{c}_k, \mathbf{c}_{k^*})$	$\frac{1}{K} \sum_{k=1}^{K} \frac{\max_{k \neq k'} Intra(k,k')}{\min_{k \neq k^*} Inter(k,k^*)}$
`GD`	$\max \frac{\mathcal{J}_k}{n_k}$	$\min_{k \neq k'} d(\mathbf{c}_k, \mathbf{c}_{k'})$	$\frac{2 \times Intra}{Inter}$
`KCE`	$\mathcal{J}^2$	1	$K \times Intra$
`PBM`	$\mathcal{J}$	$\sum_{i=1}^{N} d(\mathbf{x}_i, \mathbf{m}) \times \max_{k \neq k'} d(\mathbf{c}_k, \mathbf{c}_{k'})$	$\left(\frac{K \times Intra}{Inter}\right)^2$
`RT`	$\frac{1}{N}\mathcal{J}$	$\min_{k \neq k'} d(\mathbf{c}_k, \mathbf{c}_{k'})$	$\frac{Intra}{Inter}$
`SIL`	See text	See text	$\frac{1}{N} \sum_{i=1}^{N} \frac{Inter(\mathbf{x}_i) - Intra(\mathbf{x}_i)}{\max(Intra(\mathbf{x}_i), Inter(\mathbf{x}_i))}$
`WB`	$\mathcal{J}^2$	$\sum_{k=1}^{K} n_k d(\mathbf{c}_k, \mathbf{m})^2$	$K \times \frac{Intra}{Inter}$
`WG`	$d(\mathbf{x}_i, \mathbf{c}_k)$	$\min_{k \neq k'} d(\mathbf{x}_i, \mathbf{c}_{k'})$	$\sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \frac{Intra(\mathbf{x}_i)}{Inter(\mathbf{x}_i)}$

3 Experiments and Results

Eight synthetic two dimensional data sets coinciding with our previous study were selected[12]. Experiment were performed using MATLAB (R2018B, 64-BIT) and the same algorithm settings were used in clustering as in Niemelä et al. [13]: removing data components completely at random, discarding fully incomplete observations, minmax-scaling data to a range $[-1, 1]$, performing initialization in an iterative manner, using previously selected prototypes with K-means++ initialization algorithm, ranging K from 2 to 20, using 100 replicates in each clustering, and selecting final solutions as the lowest clustering error for the each value of K. Mean vectors and covariance matrices of incomplete multivariate normal data were estimated using `ecmnmle` method which was provided in MATLAB's Financial Toolbox.

Table 2 presents median calculation times and root mean square errors when clustering was performed with (EED, second row of results in each cell of Table 2) and without (ADS, first row of results in each cell of Table 2) distance estimation for all synthetic data sets. The clustering and missing

[1] `http://cs.uef.fi/sipu/datasets/`

[2] `http://users.jyu.fi/~mapeniem/CVI/Data/`

Table 2 The median calculation times and the obtained root mean square errors after repeated clustering. The numbers of observations are given in the brackets in the second row of the table.

ADS **EED**	**S1** (5000)	**S2** (5000)	**S3** (5000)	**S4** (5000)	**S2D2** (2000)	**S5D2** (2970)	**O200** (200)	**O2000** (2000)
Time(s)*+	12.670	15.520	21.030	23.270	1.090	4.090	1.470	5.030
	14.460	17.440	18.410	22.900	0.890	3.140	1.080	2.330
SD(s)*+	2.100	2.300	2.920	3.060	0.140	0.670	0.120	0.610
	2.203	1.973	2.054	4.544	0.067	0.389	0.238	0.198
RMSE	0.005	0.006	0.013	0.013	0.049	0.073	0.054	0.034
	0.002	0.002	0.004	0.004	0.024	0.017	0.028	0.006

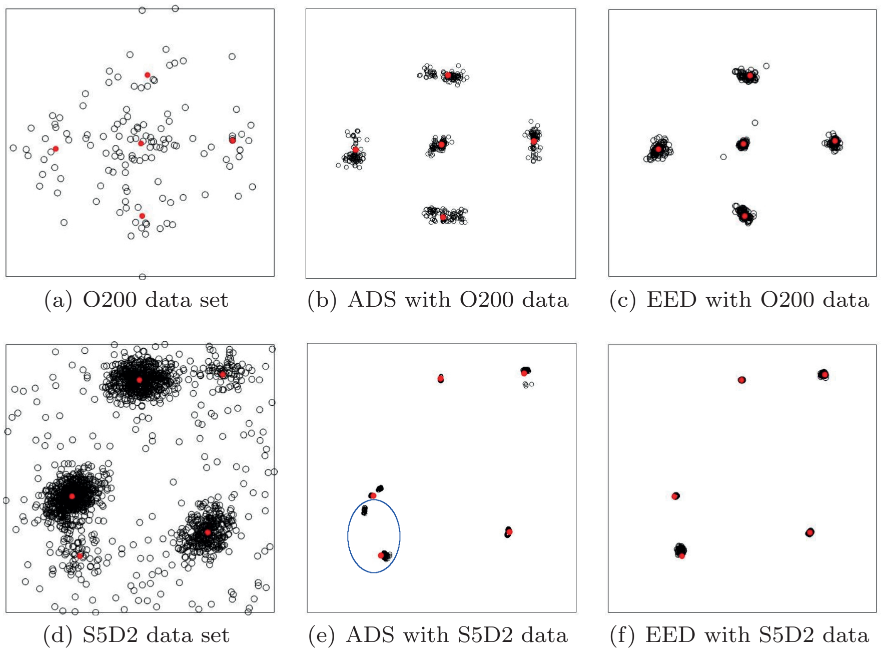
* By INTEL(R) XEON(R) CPU E5-2690 V4 @ 2.60GHZ processor without parallelization

+ Times were measured through 100 replicates in clustering

values generation were repeated 100 times using 20 % of missing values in the data. The correct numbers of clusters were used in every repetitions. The root mean square errors were calculated between the real centroids and the obtained clustering results. Regarding to the errors, the EED method provided better results with all data sets, especially with the *S5D2* and *O2000.* In addition, the EED showed almost the same computational complexity as the traditional ADS with the largest *S1–S4* data sets and to be faster with the rest of data sets.

Figure 1 shows clustering results through 100 repetitions for *O200* and *S5D2* data sets which consisted of 20 % re-generated missing values in each repetition. The obtained cluster prototypes are illustrated with the black circles. The original data centroids are visualized with the filled red circles. It can be seen from the figure that the variances of the clustered prototypes are smaller around the real prototypes when the EED distance estimation strategy was used. Further, Figure 1(e) shows some of the prototypes which were obtained with the ADS and should belong to the sparse bottom left cluster. However, these prototypes appeared to move towards to the dense cluster next to it. This is illustrated with an ellipse around prototypes.

The distance estimation strategy globally utilizes information on Gaussian distributed data while it makes decision of prototype locations and thus it appears to offer more stable results in the cases of sparse data sets. However, since the method is based on approximated quantities of the normal distributions, it can lead to nonoptimal solution locally, whereas the traditional ADS based clustering can be mathematically proofed to find a local minimum of an error function (Äyrämö [2]). This is the reason why we ended up using a two-stage clustering approach: distance estimation based clustering method first offers a high-quality initialization for the robust traditional method. The whole procedure is given in Algorithm 1. The new method was com-



(a) O200 data set (b) ADS with O200 data (c) EED with O200 data

(d) S5D2 data set (e) ADS with S5D2 data (f) EED with S5D2 data

Fig. 1 Clustering results of repeated clustering for two synthetic data sets using spatial median with and without distance estimation. The data sets consisted of 20 % missing values.

Algorithm 1 Spatial median clustering based on distance estimation

```
Input: Data set $\mathbf{X}_m$ with missing values and the number of clusters $K$
  Select initial prototypes in an iterative manner by using previously
    selected prototypes and K-means++ algorithm.
  Calculate a mean vector and a covariance matrix of the $\mathbf{X}_m$.
  repeat
    1. Estimate distances between observations and prototypes by Eq. (3).
    2. Assign individual observation to the closest prototype.
    3. Recompute prototypes with the assigned observations.
  until The final convergence
  Repeat steps 2 and 3 without distance estimation.
Output: $K$ partitions and prototypes of the given data set
```

pared against spatial median without distance estimation in the experiments related to the cluster validation.

Table 3 summarizes the results of the cluster validation indices. According to the table, the two-stage clustering approach notably improved the performance of most of the indices. Especially, the results improved in the cases of *O200* and *S5D2* data sets which were the most demanding for the indices. `Calinski-Harabasz` was the best performing index which always recommended the correct numbers of clusters with the new approach. The results of the `Calinski-Harabasz` were promising also without distance estimation since only in two out of 32 cases the index did not recommend the correct solutions. Other well performing indices were `kCE-index`, `Ray-Turi`, and `Silhouette` which recommended very often the correct numbers of clusters over all test cases.

The indices were implemented to use the ESD or EED distance estimation strategy. The strategy was selected based on the squared (ESD) or non-squared (EED) index formula (see Table 1). However, the distance estimation decreased the performance of most of the indices. Only `Silhouette` and `Wemmert-Gançarski` benefited from the estimation. Against other indices, `Silhouette` and `Wemmert-Gançarski` calculate *Inter* using distances between observations and their neighboring centroids or clusters (see Table 1 and Eq. (4)). Hence, distances were needed to be calculated more accurately for these two indices which is a good reason why the performance gain was obtained. Since distance estimation offered only marginal benefit with these two indices, we do not report results here.

4 Discussion

Let us briefly reflect the obtained results to the results of our previous study in Niemelä et al. [13]. The performance increased with most of the indices only by changing the clustering method to use the robust spatial median. The new estimation strategies yielded to performance gain. In eight over ten cases the results were at least equal, and in most of those (seven cases) better compared to the results of K-means with the partial distance strategy. However, `Wemmert-Gançarski`, which was the best performing index in Niemelä et al. [13], benefited the least from the current changes. Also, the results of `Pakhira-Bandyopadhyay-Maulik` were not improved, whereas especially `Calinski-Harabasz` and `Ray-Turi` were improved to recommend more often the correct number of clusters. The partial distance strategy was tested also in the current study but we noticed that the ADS performed better with the spatial median and, therefore, the results of the strategy were not reported here.

The new clustering method did not increase the computational complexity of the clustering. More specify, data vectors and variances were needed to be estimated only once for each observation which consisted missing values. This was done before the local refinement step of the prototype-based clustering (see Algorithm 1). Surprisingly, the calculation times were even smaller compared to the traditional spatial median clustering in cases of small data sets. However, all the data sets were only two dimensional and, hence, provided minimal challenge for the EED. In comparison, we tested the distance estimation through the whole clustering process such as estimations were repeated every time when cluster partitions were changed, i.e., as many times as final convergence was reached for each replicate of the clustering. As expected, this approach was computationally very intensive. Further, the performance of the indices did not improve as much that the method could be recommended to the clustering.

5 Conclusions

In this study, the internal cluster validation indices were compared to evaluate the number of clusters with data sets which included various ratios of missing values. The study differentiated from Niemelä et al. [13] by using similar experimental settings but extending the clustering method for more robust spatial median and utilizing the recently presented EED distance estimation strategy for clustering. The ESD and EED strategies were tested to implement to the actual indices. However, the most of the indices performed better without estimation. Thus, these results were not reported.

The study presented the new approach which performed clustering by using two stage clustering process where data sets were first clustered by using EED and, thereafter, the results were given as a starting point to the traditional ADS based spatial median clustering. On average, the new method improved the performance of the tested indices compared to the traditional ADS without distance estimation. Improved results were especially obtained when the data sets included 20 % of missing values. The best performing index was `Calinski-Harabasz`, which together distance estimation based clustering approach proposed always the correct number of clusters. The very promising results were also proposed by `kCE-index`, `Silhouette`, and `Ray-Turi` indices.

As it is well known, characteristics of real world data is rarely obvious. Therefore, it will be interesting to test the new method and the best indices with multiple of real world data sets. The special interest would be to measure the stability of indices against different ratios of missing values when the correct number of clusters is not clear.

Acknowledgements The authors would like to thank the Academy of Finland for the financial support (grants 311877 and 315550).

References

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035. SIAM, 2007.

[2] S. Äyrämö. *Knowledge mining using robust clustering.* PhD thesis, University of Jyväskylä, 2006.

[3] E. Eirola, G. Doquire, M. Verleysen, and A. Lendasse. Distance estimation in numerical data sets with missing values. *Inform. Sci.*, 240: 115–128, 2013.

[4] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki. Mixture of Gaussians for distance estimation with missing data. *Neurocomput.*, 131:32–42, 2014.

[5] W. Fu and P. O Perry. Estimating the number of clusters using cross-validation. *J. Comput. Graph. Stat.*, 29(1):162–173, 2020.

[6] J. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971.

[7] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen. Comparison of internal clustering validation indices for prototype-based clustering. *Algorithms*, 10(3), 2017.

[8] J. Hämäläinen, T. Kärkkäinen, and T. Rossi. Scalable robust clustering method for large and sparse data. In *Proceedings of ESANN2018 – 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 449–454. ESANN, 2018.

[9] T. Kärkkäinen and E. Heikkola. Robust formulations for training multilayer perceptrons. *Neural Comput.*, 16(4):837–862, 2004.

[10] T. Kärkkäinen and J. Toivanen. Building blocks for odd-even multigrid with applications to reduced systems. *J. Comput. Appl. Math.*, 131(1–2): 15–33, 2001.

[11] W.-C. Lin and C.-F. Tsai. Missing value imputation: A review and analysis of the literature (2006–2017). *Artific. Intell. Rev.*, 53(2):1487–1509, 2020.

[12] D. P. P. Mesquita, J. P. P. Gomes, A. H. Souza Junior, and J. S. Nobre. Euclidean distance estimation in incomplete datasets. *Neurocomput.*, 248:11–18, 2017.

[13] M. Niemelä, S. Äyrämö, and T. Kärkkäinen. Comparison of cluster validation indices with missing data. In *Proceedings of ESANN2018 – 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 461–466. ESANN, 2018.

[14] M. Rouaud. *Probability, Statistics and Estimation: Propagation of Uncertainties in Experimental Measurement.* 2013.

Table 3 The determined number of clusters by internal cluster validation indices. The bolded numbers indicate correct solutions. Each column correspond different percentage (0, 5, 10, and 20 %) of missing values.

ADS **EED(**)**	**CH**	**DB**	**DB***	**GD**	**KCE**
S1	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**
S2	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**
S3	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	4 **15 15 15** 4 **15 15 15**	**15 15 15 15** **15 15 15 15**
S4	**15 15 15 15** **15 15 15 15**	17 17 17 **15** 17 **15 15 15**	13 13 13 13 13 13 14 13	4 3 3 4 4 3 3 4	**15 15 15 15** **15 15 15 15**
S2D2	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**
S5D2	**5 5 5** 4(*) **5 5 5 5**	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	**5 5 5** 4(*) **5 5 5 5**
O200	**5 5 5** 20 **5 5 5 5**	**5 5 5** 20 **5 5 5 5**	**5 5 5** 20(*) **5 5 5 5**	4 4 **5 5** 4 **5** 4 **5**	**5 5** 20 20 **5 5** 17 **5**
O2000	**5 5 5 5** **5 5 5 5**	**5 5 5 5** **5 5 5 5**	**5 5 5 5** **5 5 5 5**	4 4 4 **5** 4 4 **5** 4	**5 5** 6 **5** **5 5** 6 6
Total	**8 8 8 6** **8 8 8 8**	**6 6 6 6** **6 7 7 7**	**6 6 6 5** **6 6 6 6**	**3 4 5 6** **3 5 5 5**	**8 8 6 6** **8 8 6 7**

ADS **EED(**)**	**PBM**	**RT**	**SIL**	**WB**	**WG**
S1	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**
S2	**15 15 15 15** **15 15 15 15**	**15 15 15** 14(*) **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**
S3	4 4 4 4 4 4 4 4	**15 15 15 15** **15 15 15 15**	**15 15 15** 2 **15 15 15** 2	**15 15 15 15** **15 15 15 15**	**15 15 15 15** **15 15 15 15**
S4	5 5 4 4 5 5 5 4	**15 15 15** 13 **15 15 15** 14	**15** 14 **15** 14 **15 15 15 15**	**15 15 15 15** **15 15 15 15**	17 16 17 16 17 16 16 **15**
S2D2	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**	**2 2 2 2** **2 2 2 2**	12 12 9 15 12 12 8 9	**2 2 2 2** **2 2 2 2**
S5D2	**5 5 5** 4(*) **5 5 5 5**	3 3 3 3 3 3 3 3	3 3 3 3 3 3 3 3	**5 5 5** 6(*) **5 5 5 5**	3 3 3 3 3 3 3 3
O200	**5** 3 4 4 **5** 3 4 3	**5 5 5 5** **5 5** 4 **5**	**5 5 5 5** **5 5 5 5**	19 19 20 20 19 19 17 20	**5 5** 20 20 **5 5** 20 20
O2000	3 4 4 4 4 4 3 4	**5 5 5 5** **5 5 5** 4	**5 5 5 5** **5 5 5 5**	**5 5 5 5** **5 5 5 5**	**5 5 5 5** **5 5 5 5**
Total	**5 4 4 3** **5 4 4 4**	**7 7 7 5** **7 7 6 5**	**7 6 7 5** **7 7 7 6**	**6 6 6 5** **6 6 6 6**	**6 6 5 5** **6 6 5 6**

(*) Correct result was found using the known centers as initial prototypes
(**) Uses EED distance estimation in the first stage of clustering

IV

TOOLBOX FOR DISTANCE ESTIMATION AND CLUSTER VALIDATION ON DATA WITH MISSING VALUES

by

Marko Niemelä, Sami Äyrämö, and Tommi Kärkkäinen 2021

IEEE Access, 10:352–367

Reproduced with kind permission of IEEE.

https://doi.org/10.1109/ACCESS.2021.3136435

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 5, 2021, accepted December 11, 2021, date of publication December 20, 2021,
date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136435

Toolbox for Distance Estimation and Cluster Validation on Data With Missing Values

MARKO NIEMELÄ, SAMI ÄYRÄMÖ, AND TOMMI KÄRKKÄINEN, (Senior Member, IEEE)
Faculty of Information Technology, University of Jyvaskyla, 40640 Jyvaskyla, Finland

Corresponding author: Marko Niemelä (marko.p.niemela@jyu.fi)

The work of Tommi Kärkkäinen was supported by a grant from the Academy of Finland under Grant 311877 (Demo) and Grant 315550 (HNP-AI).

ABSTRACT Missing data are unavoidable in the real-world application of unsupervised machine learning, and their nonoptimal processing may decrease the quality of data-driven models. Imputation is a common remedy for missing values, but directly estimating expected distances have also emerged. Because treatment of missing values is rarely considered in clustering related tasks and distance metrics have a central role both in clustering and cluster validation, we developed a new toolbox that provides a wide range of algorithms for data preprocessing, distance estimation, clustering, and cluster validation in the presence of missing values. All these are core elements in any comprehensive cluster analysis methodology. We describe the methodological background of the implemented algorithms and present multiple illustrations of their use. The experiments include validating distance estimation methods against selected reference methods and demonstrating the performance of internal cluster validation indices. The experimental results demonstrate the general usability of the toolbox for the straightforward realization of alternate data processing pipelines. Source code, data sets, results, and example macros are available on GitHub. https://github.com/markoniem/nanclustering_toolbox

INDEX TERMS Missing values, distance estimation, clustering, cluster validation.

I. INTRODUCTION

In many machine learning tasks, the volume of data is limited, necessitating that all the available data values be utilized as extensively as possible. The assumption that the data is complete is often invalid in real-world applications [1]. A simple strategy for avoiding the problem of missing data is to omit incomplete observations. However, this is not an efficient use of data because the important information may be lost. A more sophisticated strategy is to impute missing values as part of a data preprocessing step. Different imputation mechanisms have been developed for various data types, e.g., binary, ordinal, categorical, and string attributes [2]. The nearest neighbors method is a common imputation approach for numerical values, which uses an average (with or without weights) of the k-nearest neighbors [2].

Estimating distances is an alternative way to address problems with missing values. A well-known distance estimation method is the partial distance strategy (PDS) [3], which is also known as a general similarity measure [4]. This approach involves similar limitations as the nearest neighbors method so that its accuracy is highly correlated to the number of missing values in data. In [5] and in [6], the expected distance estimations were reported to be more accurate than the data imputation or the PDS for selected real-world data sets. However, the performance of these methods has not been tested in unsupervised machine learning tasks such as data clustering. Clustering can benefit from accurate distance estimation with missing values because both currently popular initialization methods like K-means++ [7] and the computation of cluster centroids are based on distances and not on observations themselves. In [5] and in [6], data values were assumed to be missing at random (MAR), where missingness may depend on the available data. MAR is a less restrictive mechanism than missing completely at random (MCAR) in which the values are missing independently of any other data values.

Many unsupervised and supervised techniques, and their combinations, have been used for data imputation. Imputation of missing sensor spatially and/or temporally dependent data using autoencoder and alternation projection onto convex sets

The associate editor coordinating the review of this manuscript and approving it for publication was Xi Peng.

based training was proposed in [8]. Shallow neural networks (both multi-layered perceptron and radial basis function) with genetic algorithms were put forward in [9]. Fuzzy clustering and support vector regression, also with a genetic algorithm-based parameter estimation, were hybridized in [10]. Decision trees and their ensembles were applied in [11]. More recently, deep learning methods, especially deep autoencoders, have been proposed and tested for mainly spatio-temporal data, e.g., in [12]–[18]. We do not address these more complex techniques here because of laborious tuning of an extensive number of metalevel parameters (e.g., what network architecture, how many layers, what kind of layers, which loss function, what training method, how much and what kind of data needed, etc., see [13], [19]).

Cluster analysis is often considered as one of the core techniques in descriptive data mining and knowledge discovery [20], statistics [21], and pattern recognition [22]. It is a stepwise process with at least nine elements to be chosen/carried out before achieving the results [23]–[25]. The elements are related to data selection, data preprocessing, selection of distance measure, choice of clustering criterion, selection missing data strategy, validation of the created algorithms, selection of the number of clusters, and finally, interpretation of results.

Clustering divides data into disjoint groups (clusters) where an ideal cluster is compact and isolated [24]. Partitional clustering methods use prototype points to represent clusters and, therefore, are also referred to as prototype-based clustering methods [26]. The methods are aimed to minimize the variance around the prototype points based on an error (score) function, and they are also called variance minimization techniques [27]. The iterative relocation procedure decreases the values of the error function until final convergence is reached [28], [29].

Cluster validation is a crucial part of cluster analysis, in which a clustering solution's quality (ideality) is being assessed. Cluster validation indices (CVIs) provide quality measures that indicate the number of clusters. The three main types of indices are relative, external, and internal [30]. The relative index compares multiple clustering results obtained with different initial settings of the clustering algorithm, whereas an external index utilizes additional information or metadata that can explain the number or form of the cluster structures. The external indices can be used, e.g., for comparing different clustering methods using the metadata of the actual cluster labels. However, internal cluster validation indices are probably the most commonly used estimates because they utilize only the information obtainable from data and clustering results. Numerous different clustering methods, including internal cluster validation indices, have been developed because of the high diversity of data [24]; for example, challenging data sets may include noise, overlapped clusters, multiple dimensions, and/or different densities [31].

This paper introduces a toolbox that encapsulates many methods and algorithms to perform cluster analysis in the presence of missing data. The versatile functionality allows a toolbox user to generate many forms of experimental settings and to realize various forms of new experiments to better understand and improve unsupervised learning with missing values.

The methodological bases in Sections II–V explain background theory related to distance computation with missing values, data preprocessing, clustering, and cluster validation. Section VI gives an overview of the toolbox, including descriptions of the sample data sets and essential toolbox functions. Section VII describes experiments that are divided into three parts. In the first part, the performance of distance estimation algorithms is measured in the direct estimation of pairwise distances in data sets with missing values. The second part compares clustering methods and cluster validation indices on two-dimensional (2D) data sets with missing values. In addition, the validation results, which are based on a key point selection function [32], are given. The results are validated against the reference results given in previously published research papers. In the third part, experiments are conducted on multidimensional data sets that were created by a recently published data generator [33]. Finally, the content and the toolbox performance are discussed and summarized in Sections VIII–IX.

II. COMPUTATION OF DISTANCES WITH MISSING VALUES

Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^n$ for all i, denote the observational data set with N observations of size n.

A. AVAILABLE DATA STRATEGY

The available data strategy (ADS) (see [34]) restricts distance computations to available values via binary projection vectors, $\{\mathbf{p}_i\}_{i=1}^N$, $\mathbf{p}_i \in \{0, 1\}^n$, which represent the sparsity pattern of each observation:

$$(\mathbf{p}_k)_i = \begin{cases} 1, & \text{if } (\mathbf{x}_k)_i \text{ exists,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The ADS is used in K-spatialmedians clustering (see, e.g. [35]), and it generalizes easily for various distance measures. For instance, the Euclidean distance between two incomplete n-dimensional column vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is defined as

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{i=1}^{n} ((\mathbf{p}_1)_i(\mathbf{x}_1)_i - (\mathbf{p}_2)_i(\mathbf{x}_2)_i)^2}. \tag{2}$$

B. PARTIAL DISTANCE STRATEGY

The PDS computes the sum of pairwise available vector values and scales the sum by the ratio of the original dimension of the vectors and the number of available pairwise values [4]. The Euclidean distance reads then as

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\frac{n}{n^*} \sum_{i=1}^{n} ((\mathbf{p}_1)_i(\mathbf{x}_1)_i - (\mathbf{p}_2)_i(\mathbf{x}_2)_i)^2}, \tag{3}$$

where n^* is the number of pairwise known values. Similarly to the ADS, the PDS can be generalized to other distance measures, such as the City block distance (see [36]).

C. EXPECTED SQUARED EUCLIDEAN DISTANCE

The framework for estimating the expected distance between two data vectors is presented in [5]. The proposed framework was designed for estimating squared Euclidean distances in the presence of missing data values and under the assumption of multivariate normal distribution. The assumption of multivariate normally distributed data is used for estimating expected values to replace the missing values in the data. The central limit theorem states that normal distribution can be used to approximate nearly any continuous distribution with a sufficiently large sample (see, e.g., [37]). The basic elements of the framework are given in Appendix A, and a more detailed description is given in [5].

Let us define the index sets of missing M_i and available A_i values of observation $\mathbf{x}_i$ as specified by $\mathbf{p}_i$, i.e., $M_i = \{1 \leq j \leq n | (\mathbf{p}_i)_j = 0\}$ and $A_i = \{1 \leq j \leq n | (\mathbf{p}_i)_j = 1\}$. Following the assumption that missing values are generated from conditional multivariate normal distribution, in which data values are independent, and missing values depend on the available values under the MAR assumption on the sparsity pattern, the expectation of the squared distance between two data vectors reads as:

$$E\left[||\mathbf{x}_1 - \mathbf{x}_2||^2\right] = \sum_{i=1}^{n} \left(((\mathbf{x}'_1)_i - (\mathbf{x}'_2)_i)^2 + (\sigma'_1)_i^2 + (\sigma'_2)_i^2\right),$$

$$(\mathbf{x}'_k)_i = \begin{cases} (\mathbf{x}_k)_i, & \text{if } i \in A_k; \\ E[(\mathbf{x}_k)_i | (\mathbf{x}_k)_{A_k}], & \text{if } i \in M_k; \end{cases}$$

$$(\sigma'_k)_i^2 = \begin{cases} 0, & \text{if } i \in A_k; \\ Var[(\mathbf{x}_k)_i \,|\, (\mathbf{x}_k)_{A_k}], & \text{if } i \in M_k. \end{cases} \tag{4}$$

With the complete derivation given in Appendix B, the ith observation concerning the missing values is normally distributed with the mean vector

$$(\boldsymbol{\mu}'_i)_{M_i} = (\boldsymbol{\mu})_{M_i} + \Sigma_{M_i A_i} \Sigma_{A_i A_i}^{-1}((\mathbf{x}_i)_{A_i} - (\boldsymbol{\mu})_{A_i}), \tag{5}$$

and covariance matrix

$$\Sigma'_{M_i M_i} = \Sigma_{M_i M_i} - \Sigma_{M_i A_i} \Sigma_{A_i A_i}^{-1} \Sigma_{A_i M_i}. \tag{6}$$

Estimating $\boldsymbol{\mu}$ and Σ for incomplete data is not a simple task, especially if the number of missing values is large compared to the number of available ones. A method based on available data is a fast alternative for estimating the covariance matrix [38]. However, the iterative expectation maximization (EM) algorithm with the maximum negative log-likelihood convergence criterion is more commonly used, e.g., in [5], [39], and [6].

1) EXPECTATION MAXIMIZATION

The EM is an iterative method to find the best estimates for the parameters in a statistical model [40], [41]. It consists of two alternating steps: expectation and maximization. The expectation step estimates the missing values in the data set. The maximization step optimizes the model parameters to fit the data best. The steps are repeated until the final convergence is reached.

The EM algorithm for estimating the mean vector $\boldsymbol{\mu}$ and the covariance matrix Σ of a data set with missing values under the assumption of the conditional multivariate normal distribution is given in Algorithm 1. The algorithm includes a bias matrix $\mathbf{B}$ with the same size as the covariance matrix Σ.

Algorithm 1 Expectation Maximization

```
Input: An incomplete data set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{x}_i \in \mathbb{R}^n$.
1. Compute mean vector $\boldsymbol{\mu}$ of available values of the data set.
2. Impute missing values by $\boldsymbol{\mu}$ to obtain the imputed matrix $\mathbf{X}_{imp}$.
3. Recompute $\boldsymbol{\mu}$ and compute covariance matrix $\Sigma$ by using imputed data.
4. Create a zero matrix $\mathbf{B}$ which size is equal to $\Sigma$.
until final convergence do
  for each $\mathbf{x}_i$ for which $M_i$ is nonempty do
    5. Impute missing values by using the formula (5).
    6. Use formula (6) and compute $\mathbf{B}_{M_i M_i} = \mathbf{B}_{M_i M_i} + \Sigma'_{M_i M_i}$.
  7. Recompute $\boldsymbol{\mu}$ and update covariance as $\Sigma = \Sigma + \mathbf{B}/N$.
  8. Remove the imputed values from the $\mathbf{X}$.
  9. Restore zeros to the matrix $\mathbf{B}$.
Output: Mean vector $\boldsymbol{\mu}$ and covariance matrix $\Sigma$.
```

The termination criterion for Algorithm 1 is based on the negative log-likelihood function that for the multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ reads as:

$$\begin{aligned} &\ln(\mathcal{L}(\boldsymbol{\mu}, \Sigma)) \\ &= \sum_{i=1}^{N} \frac{1}{2}[\ln(\det(\Sigma)) + (\mathbf{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) + n \ln(2\pi)] \\ &= \frac{1}{2} N[\ln(\det(\mathbf{L})^2) + n \ln(2\pi)] \\ &\quad + \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}) \\ &= \frac{1}{2} N[2 \sum_{j=1}^{n} \ln(\mathbf{L}_{jj}) + n \ln(2\pi)] \\ &\quad + \frac{1}{2} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}_i - \boldsymbol{\mu}), \end{aligned} \tag{7}$$

where $\mathbf{L}$ is obtained from the Cholesky decomposition of the covariance matrix Σ, i.e., $\Sigma = \mathbf{L}\mathbf{L}^T$. Convergence is reached when there is no significant change in the values of the log-likelihood function between successive iterations.

2) FINAL ALGORITHM

The steps for computing ESDs for incomplete data are given in Algorithm 2.

D. EXPECTED EUCLIDEAN DISTANCE

In [6], the work in [5] and [39] was continued by extending the ESD distance for the expected Euclidean distance (EED). It was shown that the EED distance could be modeled with a

```
Algorithm 2 Expected Squared Euclidean Distances
  Input: An incomplete data set $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^n$.
  1. Compute the mean vector $\boldsymbol{\mu}$ and covariance matrix $\Sigma$ of incomplete
     data set using Algorithm 1.
  for each $\mathbf{x}_i$ for which $M_i$ is nonempty do
      2. Compute the conditional mean $(\boldsymbol{\mu_i}')_{M_i}$ using formula (5) and the
         conditional covariance matrix $\boldsymbol{\Sigma}'_{M_iM_i}$ using formula (6), respectively.
      3. Impute missing values of $\mathbf{x}_i$ by values from $(\boldsymbol{\mu_i}')_{M_i}$ to obtain $\mathbf{x}'_i$.
      4. Impute conditional variance terms of $\boldsymbol{\sigma}'^{2_i}$ from the diagonal of
         $\Sigma'_{M_iM_i}$.
  for each pair of $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\{\mathbf{x}_i\}_{i=1}^N$ and in $\{\mathbf{x}_j\}_{j=i+1}^N$ do
      5. Compute the expected distance by utilizing the formula (4).
  Output: Pairwise squared Euclidean distances $\boldsymbol{d}_{ij}$ of data vectors.
```

Nakagami distribution if the distances are assumed to follow the Gamma distribution. The expected Nakagami distributed values can then be obtained as follows:

$$E\left[(\sum_{i=1}^{n}((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^2)^{\frac{1}{2}}\right]=E[z^{\frac{1}{2}}]=\frac{\Gamma(m+\frac{1}{2})}{\Gamma(m)}\left(\frac{\Omega}{m}\right)^{\frac{1}{2}},$$
$$m=\frac{E[z]^2}{Var[z]},\quad \Omega=E[z], \tag{8}$$

where m and Ω are the shape and spread parameters of the Nakagami distribution, respectively, and Γ is the Gamma function.

Under the independence assumption (as in [5], [39]), the variance can be expressed as

$$\begin{aligned}Var[z] &= Var\left[\sum_{i=1}^{n}((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^2\right]\\ &=\sum_{i=1}^{n}Var[((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^2]\\ &=\sum_{i=1}^{n}E[((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^4]-E[((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^2]^2\\ &=\Big(\sum_{i=1}^{n}E[(\mathbf{x}_1)_i^4+(\mathbf{x}_2)_i^4-4(\mathbf{x}_1)_i^3(\mathbf{x}_2)_i\\ &\quad -4(\mathbf{x}_1)_i(\mathbf{x}_2)_i^3+6(\mathbf{x}_1)_i^2(\mathbf{x}_2)_i^2]\Big)\\ &\quad -\sum_{i=1}^{n}E[((\mathbf{x}_1)_i-(\mathbf{x}_2)_i)^2]^2,\end{aligned} \tag{9}$$

where the expected values are obtainable using non-central moments. Table 1 presents moments of the normal distribution that can be used directly in the case of multivariate Gaussian distribution. However, weighted moments are needed if the data are assumed to follow Gaussian mixture distribution (see [6] for more details).

The computation of the EED distances is based on the same framework as in Algorithm 2. However, additional steps are required which are given in Algorithm 3.

TABLE 1. Non-central moments of normal distribution.

Expected value	Non-central moment
$E[\mathbf{x}_k]$	$\hat{\mathbf{x}}_k$
$E[\mathbf{x}_k^2]$	$\hat{\mathbf{x}}_k^2+\boldsymbol{\sigma}_k^2$
$E[\mathbf{x}_k^3]$	$\hat{\mathbf{x}}_k^3+3\hat{\mathbf{x}}_k\boldsymbol{\sigma}_k^2$
$E[\mathbf{x}_k^4]$	$\hat{\mathbf{x}}_k^4+6\hat{\mathbf{x}}_k^2\boldsymbol{\sigma}_k^2+3\boldsymbol{\sigma}_k^4$

```
Algorithm 3 Expected Euclidean Distances
  Input: An incomplete data set $\{\mathbf{x}_i\}_{i=1}^N$, $\mathbf{x}_i \in \mathbb{R}^n$.
  1. Utilize Algorithm 2 to obtain the spread parameter $\Omega$ for each pair of
     data vectors.
  for each pair of $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\{\mathbf{x}_i\}_{i=1}^N$ and in $\{\mathbf{x}_j\}_{j=i+1}^N$ do
      2. Compute $Var(z)$ by using formula (9) and non-central moments
         ($E[\mathbf{x}_k]$, $E[\mathbf{x}_k^2]$, $E[\mathbf{x}_k^3]$, $E[\mathbf{x}_k^4]$) given in Table 1.
      3. Use formula (8) to obtain the shape parameter $m$ and the final
         distance $d_{ij}$.
  Output: Pairwise Euclidean distances $\boldsymbol{d}_{ij}$ of data vectors.
```

III. DATA PREPROCESSING

A. FEATURE SCALING

Feature scaling is a typical preprocessing step in data analysis. Various data types are often measured in different units, which may lead to data types with large scales dominating the other data types in data-driven models. Various feature scaling approaches have been proposed, but the most commonly used approaches are the z-score and min-max normalization.

The z-score method equalizes the data type weights by transforming each one to a zero mean and unit variance. It is obtained by a linear transformation, subtracting the mean and by dividing the standard deviation:

$$x'=\frac{x-\mu}{\sigma}=\frac{1}{\sigma}x-\frac{\mu}{\sigma}=\alpha x-\beta, \tag{10}$$

where μ and σ are the sample mean and standard deviation of the available values in the data set, respectively, and x' is the scaled value.

Min-max normalization scales data to the selected range. The range may depend on the performed task, but $[-1, 1]$ and $[0, 1]$ are probably the most common choices. The min-max formula for an arbitrary range $[a, b]$ can be written as follows:

$$x'=a+\frac{(x-\min(x))(b-a)}{\max(x)-\min(x)}, \tag{11}$$

where min and max are computed for the available values.

B. K-NEAREST NEIGHBORS IMPUTATION

The k-nearest neighbors (kNN) method is a well-known and popular approach for imputing numerical values [2]. This method can be implemented by finding the closest complete observation for an incomplete observation and imputing the missing values or taking an average of k closest observations, of which some can be partially incomplete. If the missing values of the data vector are not available in the k closest observations, the k should be increased until imputation succeeds.

In the literature, there exist many variants of kNN imputation, e.g., complete-case kNNI (CCkNNI), where a data vector with missing values is imputed by using the average value of a set of k nearest complete observations, or incomplete-case kNNI (ICkNNI), where data vectors are selected from the case library in which the eligible nearest neighbors share the same complete values as $\mathbf{x}_i$ and a missing value is available. In [42], it was suggested that up to $k = 5$ neighbors should be considered. If there are not enough neighbors, the missing value is imputed by the sample mean of all the available values for that data type. Even though nearest neighbors imputation is a straightforward approach for dealing with missing values, it can be inefficient when the number of missing values is relatively high [5].

C. LOW-RANK MATRIX COMPLETION

A low-rank solution for matrix completion is a common technique for data imputation. The low-rank matrix has a decreased number of degrees of freedom and, therefore, it makes the estimation problem of missing values practical to solve [43]. The rank minimization problem can be addressed by using convex relaxation techniques utilizing the nuclear-norm [43], [44], which yields to the minimization of the following optimization problem:

$$\min_{\widetilde{\mathbf{X}}} ||\mathbf{p}\cdot\mathbf{x} - \mathbf{p}\cdot\tilde{\mathbf{x}}||_2^2 + \lambda||\widetilde{\mathbf{X}}||_*, \tag{12}$$

where $\widetilde{\mathbf{X}}$ is the completed data matrix which will be estimated, the data vectors $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are flattened versions of the data matrices $\mathbf{X}$ and $\widetilde{\mathbf{X}}$, respectively, and $\cdot$ denotes the dot product. Moreover, the vector $\mathbf{p}$ is the flattened version of the projection matrix defined in (1), λ is the regularization parameter, and $||\cdot||_*$ denotes the nuclear norm. The optimization problem in (12) can be solved iteratively by using a soft-thresholding technique to obtain the updated data vector $\tilde{\mathbf{x}}$. The initial guess of $\tilde{\mathbf{x}}$ is given by the zero vector. Then, in the kth iteration, $\tilde{\mathbf{x}}$ is updated as follows:

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_{k-1} + (\mathbf{p}\cdot\mathbf{x} - \mathbf{p}\cdot\tilde{\mathbf{x}}_{k-1}). \tag{13}$$

After that, $\tilde{\mathbf{x}}_k$ is reshaped to a matrix form $\widetilde{\mathbf{X}}_k$, the singular value decomposition is applied to the reshaped matrix, the singular values are soft-thresholded to obtain the updated $\hat{\Sigma}_k$. The $\widetilde{\mathbf{X}}_k = \mathbf{U}\hat{\Sigma}_k\mathbf{V}^T$ is flattened to obtain the final $\tilde{\mathbf{x}}_k$ in the kth iteration. The λ is reduced by a cooling algorithm such that $\lambda_1 > \lambda_2 > \ldots > \lambda_\infty$. The final result is obtained when there is a sufficiently small relative change in the target function $||\mathbf{p}\cdot\mathbf{x} - \mathbf{p}\cdot\tilde{\mathbf{x}}||_2$ or when λ reaches the predefined tolerance.

D. TRANSFORMATION INTO SPHERICAL FORM

The prototype-based K-means and K-spatialmedians clustering methods are not intended to discover any shape clusters because the used location estimates (mean and spatial median) assume spherical symmetry. That is the difference from kernel-based methods; see, e.g., [45]. Such assumption is also inherent in the computation of Inter for cluster validation indices (see Table 2). However, the assumption that a data set contains clusters with spherical shapes can be unrealistic, making the clustering and cluster validation problems more challenging. In [32], a new approach for transforming and normalizing an arbitrarily shaped subset of data to an approximately spherical shape with a specified radius was introduced. The method is based on the notation of chains around high-density key points. The original method assumes a 2D data space. Thus, multidimensional scaling (MDS) [46] can be applied to project high-dimensional data sets into the 2D.

1) DEFINITION OF KEY POINT

The M points from $\mathbf{X}$ with relatively higher density and larger density-based distances are associated with the key points which can be determined by selecting M largest values based on the following equation:

$$\begin{aligned} p_i &= \rho_i r_i, \\ \rho_i &= (\sum_{k=1}^{4} d(\mathbf{x}_i, \mathbf{x}_{i,k}))^{-1}, \quad r_i = \min_{j:\rho_j>\rho_i} d(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{14}$$

where ρ denotes the density of $\mathbf{x}_i$ and $\mathbf{x}_{i,1}\ldots\mathbf{x}_{i,k}$ are $k = 4$ nearest neighbors of $\mathbf{x}_i$, the minimum distance from $\mathbf{x}_i$ to other points with a higher density is denoted as r_i. The method connects points in the data set using density-based distance as the connection rule. Density-based connections are created until the key points are visited. In [32], the number of key points M was suggested to be selected as $\lfloor\sqrt{N}\rfloor$.

2) DEFINITION OF CHAIN

Points that are connected to a key point form a chain. Multiple chains to one key point are allowed. Let us assume c chains. Then the chain lengths can be defined as:

$$T_c = \sum_{i=1}^{n_c-1} d(\mathbf{x}_i^{(c)}, \mathbf{x}_{i+1}^{(c)}), \tag{15}$$

where n_c is the total number of points in chain c. In data set normalization, distances are transformed into a new one as follows:

$$d^*(\mathbf{x}_i^{(c)}, \mathbf{x}_{i+1}^{(c)}) = d(\mathbf{x}_i^{(c)}, \mathbf{x}_{i+1}^{(c)})/T_c. \tag{16}$$

After normalization, the lengths of the longer chains are shortened, whereas shorter chains are lengthened, i.e., longer chains move closer to the key points, and shorter chains move away from the key points. The normalized chains can be optionally scaled to a fixed size.

IV. CLUSTERING

A. BASIC ALGORITHMS

Prototype-based clustering methods consist of two main phases: selection of initial prototypes and iterative refinement until final convergence is reached, i.e., the cluster partition does not change (see Algorithm 4). In the classical

K-means [47], MacQueen's initialization phase is combined with Lloyd's search phase [48]. In general, the initialization phase is based on the random selection of initial prototypes, which most often causes the points to be selected from the same dense region yielding a poor performance [48]. Moreover, due to the initial point selections, it is known that the basic algorithm does not guarantee a unique solution to the global minimum of the error function [24]. Finding the global minimum is an NP-hard problem because there are Stirling number of the second kind different partitions for N observations into K groups [49]. In practice, the common way to perform clustering is to repeat the algorithm with multiple restarts and to use the smallest local clustering error as a selection criterion for the final prototypes [50].

The mean of the cluster points is the statistical estimate of the cluster prototype in K-means. The method assumes that data are spherical Gaussian distributed with normally distributed noise and equal variance in each cluster. The median and the spatial median, the latter also referred to as the Fermat-Weber or Weber point, are robust estimates of location [51], whose spherical symmetric distributions are uniform and Laplace distributions, respectively. The spatial median is a multivariate generalization of the univariate median. The median and the spatial median are robust prototypes of a data distribution since they can tolerate up to 50% of incorrect data values without being disturbed. The spatial median is rotation invariant so that robustness improves as the dimension of the continuous problem space grows [49].

Algorithm 4 The Main Phases of Prototype-Based Clustering

```
Input: Data set and the number of clusters $K$.
1. Select $K$ observations as the initial prototypes.
until the partition does not change do
    2. Assign each observation to the closest prototype.
    3. Recompute the prototypes with the assigned observations.
Output: Partitions and prototypes corresponding $K$ disjoint data subsets.
```

In the general case, the clustering error function can be written as follows:

$$\mathcal{J}_k = \sum_{\mathbf{x}_i \in C_k} d(\mathbf{x}_i, \mathbf{c}_k)_p^q, \quad \mathcal{J} = \sum_{k=1}^{K} \mathcal{J}_k, \tag{17}$$

where $d(\cdot)$ is the distance computation strategy in the l_p^q space, and $\{\mathbf{c}_k\}_{k=1}^K$ is the set of cluster prototypes that minimizes locally the error function (17) and partitions the data into K disjoint subsets. $\mathcal{J}_k$ is the within-cluster error in cluster C_k, and l_p-norm to the q-th power is the distance measure corresponding to the different location estimates of the error function (see [51], [52]). Specifically, the sample mean, median, and spatial median are obtained by choosing ($p = q = 2$), ($p = q = 1$), and ($p = 2, q = 1$), respectively. The sample mean and the median are straightforward to compute, whereas the spatial median requires minimization of a non-smooth (i.e., nondifferentiable) optimization problem (see [52]) that requires more complex iterative methods to be computed [53]. For instance, the solution can be obtained efficiently by using the successive over-relaxation (SOR) method.

In the K-means++ initialization approach, the first prototype is selected as the centroid of the data set. Then, the following prototypes are selected iteratively from $\mathbf{X}$ based on the probability function obtained from previous prototype(s) $\min\ d(\mathbf{x}_i, \{\mathbf{c}\}_{k=1}^{K-1}) / \sum_{x_i \in C_k} \min\ d(\mathbf{x}_i, \{\mathbf{c}\}_{k=1}^{K-1})$. Thus, the initial prototypes are very probably selected separately. The selection procedure can also be performed incrementally [54]. It means that previously obtained $K - 1$ cluster prototypes are used as a fixed set of initial points where only one point is sampled according to the K-means++ principle. In high-dimensional problems, K-means++ may show deteriorating behavior which can be compensated by using dimension reduction techniques [33].

B. CLUSTERING BASED ON EXPECTED DISTANCES

Computing the expected distances rely on the assumption of normally distributed data. The central limit theorem suggests that the assumption is valid with many continuous data sets with appropriate sample sizes [37], [55]. However, the statistical parameters of data distribution are usually unknown, and missing values make estimating parameters more challenging. Usually, the EM algorithm can produce sufficiently accurate estimators of the unknown parameters, making the clustering task more approachable because the data characteristics are better known.

A clustering algorithm based on estimated distances was presented in [56]. The core steps of the method are shown in Algorithm 5. The algorithm skeleton is identical to the traditional clustering (see Algorithm 4) but consists of two additional steps (steps 2 and 3) that utilize distance estimation. Steps 4 and 5 are repeated with estimated distances until final convergence is reached. We noticed in the previous study that, on average (over 100 repetitions), clustering based on the distance estimation produced better initial prototypes than the clustering based on the ADS. However, giving the distance-estimated prototypes as the initial points to the K-spatialmedians based on ADS produced even more accurate solutions to the clustering tasks. Thus, step 6 was included in the developed method in Algorithm 5.

V. CLUSTER VALIDATION INDICES

Many clustering algorithms require the number of clusters as an input parameter. However, often this information is not available, and deciding the number can be challenging, especially in the case of multidimensional data, which humans cannot directly conceive. Even though there are many methods for illustrating multidimensional data, i.e., using different multidimensional visualization techniques [57] or dimension reduction techniques [58], [59], the data structure may not be obvious. Cluster validity provides a way to validate the quality of the clustering results by discovering the partition that best fits the nature of the data. Thus, because of the high diversity of data, cluster validation measures, e.g., CVIs, are

Algorithm 5 Clustering Based on EED-ADS Distance Computation

```
Input: Data set X with missing values and the number of clusters $K$.
    1. Select the spatialmedian as the first prototype of the data set.
    2. Iteratively select the initial prototypes by using
       previously selected $K-1$ prototypes and K-means++
       initialization.
    2. Compute the mean vector and covariance matrices of the data using
       the EM method.
    3. Compute the expected distances between the observations and
       prototypes.
    repeat
       4. Assign individual observations to their closest prototypes.
       5. Recompute the prototypes with the assigned observations.
    until The final convergence
    6. Repeat steps 4 and 5 without distance estimation.
Output: Partitions and prototypes corresponding $K$ disjoint data subsets.
```

recommended, even essential, methods for determining the final number of clusters [31].

A. INTERNAL CLUSTER VALIDATION INDICES

Internal cluster validation indices are commonly based on two measures: 1) Compactness, also referred to as Intra, indicates how close the observations are to each other within the same cluster. A commonly used Intra is a clustering error itself, e.g., in the `Ray-Turi` index. 2) Separability, also known as Inter, indicates how distant a cluster is from the other clusters. Typically, Inter is computed as the minimum or maximum distance between all prototypes. Variability between prototypes around the centroid of the data is also used by many indices, e.g., in the `Calinski-Harabasz` index. In general, the purpose of CVIs is to minimize Intra and to maximize Inter, so that the argument minimum or maximum of division indicates the number of clusters.

Table 2 specifies the Inters and Intras of the best internal cluster validation indices according to [56]. Indices are presented in a general fashion for l_p^q-norm settings. Explanation of abbreviations are given in Table 3. The whole data prototype is denoted by $\mathbf{m}$, whereas n_k indicates the number of observations in the kth cluster. The special distance computation strategies given in Section II, denoted by $d(\cdot)$, are required if at least one data vector includes missing values. Note that the `WB-index`, `Calinski-Harabasz`, and `kCE-index` include penalization terms for a high number of clusters that were originally derived in the context of the squared formulas. Therefore, l_p^2-norms were used for these indices regardless of the clustering error criterion used. In the `Silhouette` index, Intra is the average dissimilarity of $\mathbf{x}_i$ to all other points in the same cluster, and Inter is the minimum average dissimilarity of $\mathbf{x}_i$ to all points in a different cluster:

$$\text{Intra}(\mathbf{x}_i) = \frac{1}{n_k - 1} \sum_{\mathbf{x}_j \in C_k} d(\mathbf{x}_i, \mathbf{x}_j),$$
$$\text{Inter}(\mathbf{x}_i) = \min_{k \neq k'} \frac{1}{n_{k'}} \sum_{\mathbf{x}_j \in C_{k'}} d(\mathbf{x}_i, \mathbf{x}_j), \tag{18}$$

where $\mathbf{x}_i$ belongs to cluster C_k.

TABLE 2. Internal cluster validation indices in general fashion.

Abbr	Intra	Inter	Formula
CH	$\mathcal{J}^2$	$\sum_{k=1}^{K} n_k \lVert\mathbf{c}_k - \mathbf{m}\rVert_p^2$	$\frac{K-1}{N-K} \times \frac{\mathit{Intra}}{\mathit{Inter}}$
DB	$\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}}$	$\lVert\mathbf{c}_k - \mathbf{c}_{k^*}\rVert_p^q$	$\frac{1}{K} \sum_{k=1}^{K} \max_{k \neq k'} \frac{\mathit{Intra}(k,k')}{\mathit{Inter}(k,k')}$
DB*	$\frac{\mathcal{J}_k}{n_k} + \frac{\mathcal{J}_{k'}}{n_{k'}}$	$\lVert\mathbf{c}_k - \mathbf{c}_{k^*}\rVert_p^q$	$\frac{1}{K} \sum_{k=1}^{K} \frac{\max_{k \neq k'} \mathit{Intra}(k,k')}{\min_{k \neq k^*} \mathit{Inter}(k,k^*)}$
GD	$\max \frac{\mathcal{J}_k}{n_k}$	$\min_{k \neq k'} \lVert\mathbf{c}_k - \mathbf{c}_{k'}\rVert_p^q$	$\frac{2 \times \mathit{Intra}}{\mathit{Inter}}$
KCE	$\mathcal{J}^2$	1	$K \times \mathit{Intra}$
PBM	$\mathcal{J}$	$\max_{k \neq k'} \lVert\mathbf{c}_k - \mathbf{c}_{k'}\rVert_p^q$	$(\frac{K \times \mathit{Intra}}{\mathit{Inter}})^2$
RT	$\frac{1}{N}\mathcal{J}$	$\min_{k \neq k'} \lVert\mathbf{c}_k - \mathbf{c}_{k'}\rVert_p^q$	$\frac{\mathit{Intra}}{\mathit{Inter}}$
SIL	See text	See text	$\frac{1}{N} \sum_{i=1}^{N} \frac{\mathit{Inter}(\mathbf{x}_i) - \mathit{Intra}(\mathbf{x}_i)}{\max(\mathit{Intra}(\mathbf{x}_i), \mathit{Inter}(\mathbf{x}_i))}$
WB	$\mathcal{J}^2$	$\sum_{k=1}^{K} n_k \lVert\mathbf{c}_k - \mathbf{m}\rVert_p^2$	$K \times \frac{\mathit{Intra}}{\mathit{Inter}}$
WG	$d(\mathbf{x}_i, \mathbf{c}_k)$	$\min_{k \neq k'} d(\mathbf{x}_i, \mathbf{c}_{k'})$	$\sum_{k=1}^{K} \sum_{\mathbf{x}_i \in C_k} \frac{\mathit{Intra}(\mathbf{x}_i)}{\mathit{Inter}(\mathbf{x}_i)}$

TABLE 3. Explanations of abbreviations.

Abbreviation Name			
CH	DB	DB*	GD
Calinski-Harabasz	Davies-Bouldin	Davies-Bouldin*	Generalized Dunn
KCE	PBM	RT	SIL
kCE-index	Pakhira-Bandyopadhyay	Ray-Turi	Silhouette
WB	WG		
WB-index	Wemmert-Gançarski		

B. EXTERNAL CLUSTER VALIDATION INDICES

External cluster validation indices can validate the quality of the clustering result if the actual clustering labels are known. The simple external index is `Accuracy-index (ACC)` which computes the quotient of the correctly predicted data labels and the total number of the labels [60]. The `normalized mutual information index (NMI)` origins from information theory. The mutual information explains the reduction in the entropy between the real and the predicted cluster labels [61]. The normalization is used to scale the result to the range of [0, 1]. Many variants exist to normalize the mutual information, e.g., min, max, and square-root normalizations [61]. However, the arithmetic method is often used, which divides the mutual information by the average value of entropy terms as follows:

$$NMI = \frac{I(L_{real}, L_{pred})}{(H(L_{real}) + H(L_{pred}))/2}, \tag{19}$$

where $I(\cdot,\cdot)$ denotes the mutual information between the real and predicted clusters and $H(\cdot)$ denotes the entropy function. The `adjusted Rand index (ARI)` measures similarity between two clusterings of the same data using the permutation model [61]. The equation can be written as:

$$ARI = \frac{\Sigma_{ij}\binom{n_{ij}}{2} - [\Sigma_i\binom{a_i}{2}\Sigma_j\binom{b_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\Sigma_i\binom{a_i}{2} + \Sigma_j\binom{b_j}{2}] - [\Sigma_i\binom{a_i}{2}\Sigma_j\binom{b_j}{2}]/\binom{n}{2}}, \tag{20}$$

where n_{ij} is an intersection table between the real and predicted cluster labels, the row and column sums of the intersection table are denoted by a_i and b_j, respectively.

VI. OVERVIEW OF THE TOOLBOX

The toolbox was implemented by using MATLAB (R2018b, 64-bit), and it is freely available with the MIT License on GitHub online[1]. The toolbox contains `benchmark_data`, `toolbox`, `test_macro`, and `results` folders.

Eight real-world classification data sets were selected from the University of California at Irvine (UCI)[2] Machine Learning Repository [62]. Seven were used in the first part of the experiments, and three were used in the second part. Further, eight synthetic data sets, including the four `S` sets[3] (15 centers and 5000 observations in each set), the `Sim5D2` set[4] (5 centers and 2970 observations), the `Sim2D2` set[4] (2 centers and 2000 observations), the `O200` set[4] (5 centers and 200 observations), and the `O2000` set[4] (5 centers and 2000 observations) were selected from a previous study [56] for the second part of the experiments. These synthetic data sets are two-dimensional. In addition, in total, 12 synthetic multidimensional data sets (10D, 50D, 100D) with 15 centers and 6000 observations in each set were created with the data set generator[5] [33] for the third part.

The toolbox includes routines for handling missing values, data preprocessing, clustering, and validating clusters. All the developed methods generalize to missing values in the data. The descriptions of the most vital MATLAB functions are given in Table 4. Notice that more detailed descriptions of functions are available using the `help` command in MATLAB (see the next section for the use case examples). Further, the `help` command shows the function calls, input and output parameters, and default values of the input parameters for each toolbox function. The toolbox supports computation strategies based on available data (ADS), partial distance (PDS), and expected distances (ESD, EED) that are used by clustering methods, cluster validation indices, and data preprocessing methods. In total, ten well-performing internal cluster validation indices depicted in Section V are supported. Further, as depicted in Sections II–III, the preprocessing functionality includes routines for data imputation, distance computation with a selected distance strategy, selecting key points, and transforming data sets into spherical forms.

A. GENERAL USE OF THE TOOLBOX

General use of the toolbox is demonstrated in the `toolboxdemo` macro (see the next section). The correct functionalities of the toolbox functions can be evaluated with test macros divided into three test case folders. The first test case folder includes the `Main` macro that performs comparisons of techniques for handling missing values. The macro selects the parameters used from the `params` file. The cluster validation process can be divided into three tasks in the second test case folder: data preparation, clustering, and cluster validation. The `Main` macro pipelines these tasks to one process and outputs an Excel file of the cluster validation results. Further, the toolbox offers missing values generation, clustering, and cluster validation as separate processes implemented in the `generatemissdata`, `clusterdata`, and `validateclustdata` macros, respectively. An optionally `visualizeresults` macro can be used to visualize the final results of the clustering and cluster validation.

The third test case folder includes the same `Main` macro functionalities as given in the second test case folder. However, the mechanism for generating missing values was modified to restore 0.5% of the original observations. It was required because the initialization of clustering uses the complete observations, and removing data values completely at random from high-dimensional data causes all observations to contain missing values.

TABLE 4. Core functions.

Name	Description
normalizedata	Performs min-max or z-score normalization
genmissdata	Generates missing values to input data
knnimpute	Imputes missing values by using k-nearest neighbors
ICknnimpute	Utilizes Incomplete-Case k-nearest neighbors imputation
distancecalc	Computes distances of data with missing values
ecmnmlefunc	Obtains conditional means and variances of incomplete data
datasetmap	Transforms data approximately spherical symmetric
kcentroids	K-centroids clustering based on available data
kcentroids_partial	K-centroids clustering based on partial distances
kcentroids_expected	K-centroids clustering based on expected distances
scatter_results	Visualizes results of clustering
iterative_kcentroids	Clusters iteratively to the maximum number of clusters
keypointsComp	Computes selected number of key points
iterative_kcentroids_kp	Performs clustering iteratively based on selected key points
cluster_validation	Computes values of cluster validation indices
plot_indices	Plots curves of cluster validation indices

B. EXAMPLES OF BASIC USE

The basic use of the toolbox is given in the `toolboxdemo` file. It includes function calls for data preprocessing, clustering, and cluster validation. In the first example, 10% of missing values are generated for the input data. The result is min-max scaled to a range of [-1, 1], and the k-nearest neighbors imputation with five neighbors is performed. Then, the dimensionality of the imputed data set is reduced to 2D and transformed into a spherical form (Section III-D). Finally, the transformed data are visualized on a scatter plot.

```
load fisheriris;
X = meas;
addpath(‘../../../toolbox/preprocess’);
Xm = genmissdata(X, 0.1);
Xnorm = normalizedata(Xm, ‘min-max’, [−1, 1]);
Ximp = knnimpute(Xnorm, 5);
Xmapped = datasetmap(Ximp);
scatter_data(Xmapped);
```

In the second example, clustering is performed based on available data in distance computation, i.e., using a

[1] https://github.com/markoniem/nanclustering_toolbox

[2] https://archive.ics.uci.edu/ml/index.php

[3] http://cs.uef.fi/sipu/datasets/

[4] http://users.jyu.fi/ mapeniem/CVI/Data/

[5] https://github.com/jookriha/M_Spheres_Dataset_Generator

K-spatialmedians clustering method. The toolbox also supports clustering algorithms based on partial (kcentroids_partial) and expected (kcentroids_expected) distances. The clustered data set, the number of clusters, the number of replicates, the distance metric, the initialization criterion, and the initial values of the centroids are given as input parameters for the clustering function. The output parameters are cluster labels for each observation, the cluster centroids, and the within-cluster sums of points-to-centroid distances:

```
addpath('../../../toolbox/kcentroids');
[ L, C, sumd] = kcentroids(Xnorm, 5, 100, 'euc',
'kmeans++', [ ] );
```

In the final example, clustering is performed iteratively, with K ranging from 2 to 10. The default values of the parameters are used in clustering (see `help iterative_kcentroids`). The centroids and labels are used as input parameters for the cluster validation function. There are two ways to specify the indices (see the example). Finally, the results of the cluster validation indices are visualized:

```
addpath('../../../toolbox/cluster_indices');
% help iterative_kcentroids;
[ centers, labels] = iterative_kcentroids(Xnorm, 10);
% Select the cluster validation indices. The 'dist' parameter
% defines the selected distance metric used by inidices.
dist = 'euc';
indices = {@CalinskiHarabasz; @DaviesBouldin; @kCE;};
% An optional way to define indices. This overrides the 'dist' option.
indices = [ @CalinskiHarabasz, 'sqe'; @DaviesBouldin, 'euc'; @kCE,
'sqe'; ] ;
indices_values = cluster_validation(Xnorm, centers, labels, dist,
indices);
%
% In default, indices use available data strategy based computation.
% However, expected distances or partial distances are supported as well.
% indices_values =
%          cluster_validation(Xnorm, centers, labels, dist, indices, 'exp');
%
plot_indices(indices, indices_values);
```

VII. EXPERIMENTAL RESULTS

Experiments were divided into three parts which are discussed in the following sections.

A. VALIDATION OF DISTANCE ESTIMATION METHODS

In the first case, the experimental settings and the reference results were obtained from [5]. The real-world data sets were selected from the UCI repository. The experiments consisted of the z-score scaling of the data to the zero mean and unit variance. Then, the fixed probabilities (5, 15, 30, and 60%) of the data values were removed completely at random from each data set. The estimated distances were compared to the real distances, which were computed beforehand. The root mean square error (RMSE) between the real distances and the estimated distances was used. The RMSE included only the cases where estimations were needed, i.e., distances over complete observations were omitted. Further, in the cases of empty data vectors, the average distances over the data samples were used in error computing. The mean values and standard deviations of the results were recorded utilizing measurements over 250 repetitions.

We validated the functionalities of the implemented distance estimation algorithms against the reference methods given in [5]. An extension of the reference paper was the self-made implementation of the EM algorithm so that the `ecmnmle` function was not required (available only in MATLAB's commercial Financial Toolbox). Further, in addition to the ESD, PDS, and ICkNNI ($k = 5$) methods, the EED, ADS, kNNI ($k = 5$), and iterative soft-thresholding methods were added to the comparisons. Table 5 shows the results, which are in line with the reference results in the six cases over seven data sets. The exception is the `wine` data set, in which all distance computation mechanisms produced different results. In [5], a Monte Carlo simulation was used to remove data values in each repetition, whereas in our experiments, data values were removed completely at random. That may explain the differences in the results. In general, the results indicate that the EED is the best-performing algorithm. However, the ESD results are only slightly worse, and the method is computationally less expensive. Thus, the ESD method is highly recommended for computing pairwise distances.

B. PERFORMANCE EVALUATION OF CLUSTERING AND CLUSTER VALIDATION

In the second part, the data clustering and cluster validation indices methods were evaluated. The initial settings were selected from [56]. These settings included removing data values completely at random from data sets (see the toolbox overview section for detailed descriptions of the data sets), min-max scaling that results in a range of $[-1, 1]$, repeating the K-spatialmedians clustering with 100 replicates, and selecting the lowest local minima as the best clustering partition. The prototypes were initialized incrementally, benefiting the previous prototypes (see the last paragraph in Section IV-A). In [56], the clustering method based on the expected distances and giving the obtained prototypes as inputs in the K-spatialmedians with ADS algorithm was suggested. The clustering and cluster validation indices were revealed to be slightly more accurate based on the two-stage clustering approach. Thus, the same procedure was repeated in this study among the K-spatialmedians clustering.

The results given in [56] were reproduced to validate the functionality of the cluster validation. Note that the `results/params` folder includes the parameter files used in different experiments related to cluster validation. The experiments showed that the best cluster validation results are obtained using K-spatialmedians clustering based on EED-ADS distance estimation. The new approach improved the performance, especially when compared to the results, which were available using the real centers of the synthetic data sets as the initial points to K-spatialmedians based on the ADS (see `results` folder). The best-performing index was `Calinski-Harabasz (CH)` that always recommended

TABLE 5. The average RMSEs and standard deviations (over 250 repetitions) of distance computation algorithms in the direct estimation of pairwise distances with data sets consisting (5, 15, 30, and 60%) of missing values. The best results for each test set are underlined, and the results that are not statistically significantly different (two-tailed paired t-test, $\alpha = 0.01$) are in bold.

		EED	**ESD**	**PDS**	**ADS**	**ICkNNI ($k = 5$)**	**kNNI ($k = 5$)**	**Soft-thresholding**
Iris	5%	**0.240** (0.048)	**0.246** (0.045)	0.436 (0.049)	0.618 (0.062)	**0.244** (0.055)	0.261 (0.065)	0.294 (0.071)
$(N = 150, n = 4)$	15%	**0.321** (0.042)	**0.329** (0.039)	0.592 (0.039)	0.875 (0.055)	0.335 (0.063)	0.493 (0.088)	0.401 (0.059)
	30%	**0.481** (0.043)	0.492 (0.039)	0.839 (0.030)	1.328 (0.045)	0.516 (0.064)	0.925 (0.060)	0.612 (0.057)
	60%	**0.920** (0.035)	0.931 (0.032)	1.208 (0.026)	2.262 (0.027)	1.175 (0.069)	1.420 (0.040)	1.247 (0.071)
Ecoli	5%	**0.450** (0.254)	**0.458** (0.250)	0.728 (0.180)	0.791 (0.193)	**0.439** (0.251)	**0.444** (0.247)	**0.469** (0.253)
$(N = 336, n = 7)$	15%	**0.650** (0.238)	**0.661** (0.234)	1.057 (0.142)	1.170 (0.159)	**0.647** (0.246)	0.705 (0.235)	0.698 (0.241)
	30%	**0.960** (0.268)	**0.976** (0.259)	1.654 (0.121)	1.815 (0.142)	**0.990** (0.268)	1.142 (0.255)	1.081 (0.261)
	60%	**1.535** (0.275)	**1.563** (0.277)	2.431 (0.103)	3.004 (0.076)	1.710 (0.244)	1.774 (0.224)	1.884 (0.226)
Breast tissue	5%	**0.216** (0.099)	**0.217** (0.097)	0.428 (0.065)	0.568 (0.092)	0.244 (0.102)	**0.236** (0.103)	0.240 (0.110)
$(N = 106, n = 9)$	15%	**0.347** (0.116)	**0.349** (0.114)	0.659 (0.067)	0.972 (0.106)	0.456 (0.137)	0.418 (0.139)	0.404 (0.128)
	30%	**0.584** (0.192)	**0.590** (0.190)	1.072 (0.070)	1.640 (0.110)	0.887 (0.149)	0.758 (0.164)	0.667 (0.142)
	60%	**1.183** (0.168)	**1.197** (0.165)	2.055 (0.112)	3.086 (0.101)	1.792 (0.272)	1.736 (0.204)	1.438 (0.198)
Glass	5%	**0.238** (0.086)	**0.242** (0.084)	0.539 (0.087)	0.650 (0.103)	0.361 (0.137)	0.363 (0.135)	0.328 (0.138)
$(N = 214, n = 9)$	15%	**0.411** (0.082)	**0.420** (0.079)	0.809 (0.057)	1.055 (0.081)	0.569 (0.117)	0.564 (0.111)	0.546 (0.121)
	30%	**0.739** (0.097)	**0.755** (0.092)	1.330 (0.059)	1.758 (0.082)	0.972 (0.128)	0.962 (0.129)	0.934 (0.133)
	60%	**1.417** (0.079)	1.437 (0.073)	2.372 (0.070)	3.199 (0.057)	1.804 (0.150)	1.828 (0.128)	1.847 (0.134)
Wine	5%	**0.414** (0.227)	**0.416** (0.226)	0.613 (0.166)	0.701 (0.165)	**0.418** (0.210)	**0.421** (0.210)	**0.416** (0.212)
$(N = 178, n = 13)$	15%	**0.702** (0.299)	**0.706** (0.296)	1.024 (0.161)	1.260 (0.166)	**0.738** (0.262)	**0.731** (0.262)	0.757 (0.263)
	30%	**1.048** (0.274)	**1.053** (0.269)	1.644 (0.104)	2.107 (0.135)	1.206 (0.268)	1.163 (0.277)	1.268 (0.268)
	60%	**1.512** (0.144)	**1.519** (0.133)	2.952 (0.152)	3.762 (0.064)	2.002 (0.230)	1.776 (0.227)	2.291 (0.195)
Parkinsons	5%	**0.181** (0.048)	**0.181** (0.047)	0.335 (0.031)	0.531 (0.039)	0.253 (0.049)	0.210 (0.048)	0.212 (0.050)
$(N = 195, n = 22)$	15%	**0.318** (0.042)	**0.319** (0.042)	0.600 (0.029)	1.203 (0.050)	0.668 (0.062)	0.408 (0.053)	0.403 (0.051)
	30%	**0.511** (0.037)	**0.513** (0.036)	0.995 (0.035)	2.235 (0.067)	1.389 (0.231)	0.744 (0.075)	0.703 (0.055)
	60%	**0.970** (0.081)	**0.973** (0.081)	2.323 (0.076)	4.349 (0.076)	2.639 (0.123)	2.044 (0.180)	1.577 (0.114)
Sonar	5%	**0.191** (0.026)	**0.191** (0.026)	0.342 (0.023)	0.678 (0.031)	0.370 (0.038)	0.268 (0.038)	0.234 (0.036)
$(N = 208, n = 60)$	15%	**0.465** (0.041)	**0.466** (0.041)	0.652 (0.031)	1.793 (0.041)	1.023 (0.053)	0.614 (0.056)	0.571 (0.053)
	30%	**0.786** (0.049)	**0.787** (0.049)	1.079 (0.034)	3.460 (0.049)	2.001 (0.063)	1.142 (0.070)	1.137 (0.069)
	60%	**1.435** (0.075)	**1.436** (0.075)	2.501 (0.056)	6.826 (0.049)	4.264 (0.082)	2.229 (0.103)	2.682 (0.105)

the correct numbers of clusters even if the data sets and degrees of missing values varied. The other well-performing indices were `kCE-index (KCE)`, `Silhouette`, and `Ray-Turi`.

Three external cluster validation indices were selected to measure the quality of the K-spatialmedians clustering results based on ADS and EED-ADS distance estimations strategies. The selected indices were: Accuracy (ACC), adjusted Rand index (ARI), and normalized mutual information (NMI). Table 6 shows the comparison results. Clearly, the EED-ADS-based estimation produces better solutions for the synthetic data sets. Especially, the better results were obtained with the `S2` data set and with the challenging `S4` and `Sim5D2` data sets. These results are in line with the results obtained by the internal indices, which especially recommended the better solutions with the EED-ADS estimation for the `Sim5D2` data set.

We applied the expected distance estimation to the actual cluster validation indices. It appeared that only `Silhouette`, `Wemmert-Gançarski (WG)`, and `Davies-Bouldin (DB)` benefited from the distance estimation, and the other indices decreased the performance for finding the correct number of clusters in the data sets. Compared to the other indices, which compute the pairwise distances between observations and complete centroids, `Silhouette` computes the pairwise distances between observations, which may be incomplete (see eq. (18)). Thus, it was expected that `Silhouette` performed better using the expected distances.

Key point selection (presented in Section III-D1) was used in the cluster validation. The number of key points can be fixed to $||\sqrt{N}||$, as recommended in [32]. However, we provided two modified versions of the original algorithm based on key point pruning, i.e., the algorithms started from the given maximum for the key points and then removed irrelevant points one by one. This was performed iteratively until the value of K of the chosen number of clusters was reached. The selected points were then used in the initialization of the selected clustering algorithm in each iteration. The experiments were performed for 2D data sets. For this purpose, `Ecoli`, `Iris`, and `Seeds` real-world data were transformed to 2D using multidimensional scaling. The selection assumed that the data sets were complete, therefore, the ICkNN ($k = 2$) imputation strategy was applied to the data sets with missing values. The figures for the key point selection result are given in `results/key_point_selection/img` folder in toolbox. The results for the cluster validation indices are given in Table 7. The reference results for the synthetic data sets were obtained from [56]. On average, the validation results for the key point selection were almost the same as the reference results, which were based on available data strategy and replicated clustering. The most challenging synthetic data set was `Sim5D2`. None of the indices was able to get all correct recommendations with the different degrees of

TABLE 6. The quality of clustering results determined with external cluster validation indices. The K-spatialmedians clusterings with available data strategy (ADS) and using both expected distance and available data computations (EED-ADS) were compared. The highest scores are bolded only if they differ between the two clustering methods.

ADS	**ACC**				**ARI**				**NMI**			
EED-ADS	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%
S1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0,974	1.000	1.000	1.000	1.000
	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**1.000**	1.000	1.000	1.000	1.000
S2	1.000	0.994	0.990	0.947	1.000	0.987	0.981	0.969	1.000	0.990	0.999	0.970
	1.000	**1.000**	**1.000**	**1.000**	1.000	**1.000**	**1.000**	**0.999**	1.000	**1.000**	**1.000**	**0.999**
S3	1.000	**1.000**	0.979	0.929	1.000	**0.999**	0.943	0.909	**1.000**	0.998	**1.000**	0.931
	1.000	0.999	**0.999**	**1.000**	1.000	0.998	**0.999**	**0.999**	0.999	0.998	0.999	**0.999**
S4	0.998	0.999	0.999	0.999	0.997	0.998	0.994	0.785	0.996	0.998	0.994	0.963
	0.998	**1.000**	**1.000**	0.999	0.997	**1.000**	**0.999**	**0.998**	0.996	**1.000**	**0.998**	**1.000**
Sim2D2	1.000	0.999	1.000	1.000	1.000	0.996	1.000	1.000	1.000	0.990	1.000	1.000
	1.000	**1.000**	1.000	1.000	1.000	**1.000**	1.000	1.000	1.000	**1.000**	1.000	1.000
Sim5D2	0.999	1.000	1.000	0.796	0.999	1.000	1.000	0.748	0.997	1.000	**1.000**	0.777
	1.000	1.000	1.000	**1.000**	0.999	1.000	1.000	**1.000**	0.997	1.000	0.998	**1.000**
O200	0.990	**1.000**	**1.000**	0.979	0.976	**1.000**	**0.986**	0.972	0.969	**1.000**	**1.000**	0.947
	0.990	0.990	0.990	**0.995**	0.976	0.971	0.978	**0.985**	0.969	0.973	0.970	**0.983**
O2000	1.000	1.000	**1.000**	0.877	1.000	1.000	**1.000**	0.737	1.000	1.000	1.000	0.762
	1.000	1.000	0.999	**1.000**	1.000	1.000	0.999	**1.000**	1.000	1.000	1.000	**1.000**

missing values. The high-density clusters in `Sim5D2` caused many incorrect validation results. The sparse clusters were connected to higher-density ones after clustering, and therefore, many indices supported three as the correct number. Especially, the sparse clusters almost disappeared based on the ICkNN imputed data with 20% of missing values (see images from `results` folder). The validation results with real-world data were improved using the key point selection with all data sets. It appeared that `CH`, `KCE`, and `WB` indices recommended a very high number of clusters for `Ecoli` and `Iris` data sets without the key point selection.

C. CLUSTER VALIDATION WITH MULTIDIMENSIONAL DATA

In the third part of the experiments, the cluster validation indices were applied to multidimensional data sets that were created by the data set generator presented in [33]. The generator draws a random point on the M-dimensional sphere centered on $\mathbf{c}$ with radius d. The distance between centers is defined as $d_c = ||\mathbf{c}_i - \mathbf{c}_j||$, $\mathbf{c}_i, \mathbf{c}_j \in C$, where $i \neq j$, and C is a set of centers. The radius d is uniformly selected from the range of (0, 1] for each data point. It means the clusters do not overlap in the multidimensional space when the distance of the centers is $d_c \geq 2$, and the cluster overlap is approximately 50% if the distance is $d_c = 1$.

Table 8 shows the results of the cluster validation indices with the predefined number of missing values (0, 5, 10, and 20%) and different degrees of cluster overlap ($d_c = [0.9, 0.8, 0.7, 0.6]$). The best performing index was `WG` which recommended the correct number of clusters in almost all test cases (45/48 correct recommendations). Interestingly, the `CH`, `KCE`, and `WB-index`, which included the squared penalization term, always recommended the incorrect number of clusters. We also tested the non-squared penalizations but were not able to improve the results. The `KCE` uses only Intra which explains that the better separation in the multidimensional space depends on the quality of Inter. It supports the finding given in [33] that the difference between the clustering errors of good and bad clustering results in high-dimensional spaces is small. The curse of dimensionality can explain the findings, which causes relative differences between the distances to vanish in high dimensional spaces [63]. The other well-performing indices were `GD`, `RT`, and `DB`*, which recommended 37, 33, and 30 correct solutions, respectively. The highest overlapping clusters ($d_c = 0.6$) were challenging for the indices because only `WG` (11/12 times), `RT` (3/12 times), and `GD` (3/12 times) were able to find the correct numbers.

The experiments were also conducted with 2D-scaled `M-Sphere` data sets. However, the performance of all indices was poor in 2D data space (only a few correct recommendations), and therefore, these results were not reported. The generated clusters were compact and isolated in the high-dimensional space, which explains the far better validation results with these data sets in their original dimensions [63]. Further, the dimension reduction leads to a loss of information which also supports the findings. Nevertheless, the developed key point selection algorithms with ICkNN ($k = 2$) imputed data possess multidimensional functionality. The results of cluster validation indices with the key point selection and multidimensional `M-Sphere` data sets are given in the `results` folder in the toolbox. The indices can be concluded to perform better when the key point selection procedure was used to initialize the K-spatialmedians clustering with 0%, 5%, and 10% of missing values in the data sets. However, a decreased performance was observed with 20% of missing values in data.

VIII. DISCUSSION

The results indicate that the ESD distance estimation could be a better choice than EED in the general case due to the lower computational complexity. The overall best clustering models with synthetic 2D data sets seem to be obtained using expected distances in clustering and giving the prototypes as inputs to the K-spatialmedians originally based on the

TABLE 7. The number of clusters determined with internal cluster validation indices based on key point selection (every second row). The data sets consisted of predefined numbers of missing values, and experiments were performed using the K-spatialmedians clustering algorithm. Reference results were obtained using K-spatialmedians clustering with available data strategy.

REF	**CH**				**DB**				**DB***				**GD**				**KCE**			
KP	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%
S1	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
S2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
S3	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	4	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	4	4	4	4	**15**	**15**	**15**	**15**
S4	**15**	**15**	**15**	**15**	17	17	17	**15**	13	13	13	13	4	3	3	4	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	14	14	11	**15**	14	14	5	3	4	3	**15**	**15**	**15**	**15**
Sim2D2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	20	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
	2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
Sim5D2	**5**	**5**	**5**	4	3	3	3	3	3	3	3	3	3	3	3	3	**5**	**5**	**5**	4
	4	3	4	3	3	3	3	3	3	3	3	3	3	3	3	3	4	**5**	**5**	4
O200	**5**	**5**	**5**	**5**	**5**	**5**	**5**	18	**5**	**5**	**5**	18	4	4	**5**	**5**	**5**	**5**	20	19
	5	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	4	4	4	**5**	4	4	**5**	**5**	**5**	**5**
O2000	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	4	4	4	**5**	**5**	**5**	6	**5**
	5	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	4	4	4	**5**	**5**	**5**	**5**	**5**
Ecoli	10	8	9	20	**3**	19	20	20	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	10	8	9	20
	2	2	2	2	**3**	**3**	**3**	20	**3**	**3**	**3**	2	**3**	**3**	**3**	2	**3**	**3**	**3**	**3**
Iris	17	20	17	5	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	17	20	17	20
	3	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	15	**3**	12	**3**
Seeds	2	2	2	16	2	2	2	2	2	2	2	2	2	2	2	2	**3**	**3**	**3**	20
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	**3**	**3**	**3**	**3**
Total	**8**	**8**	**8**	**7**	**8**	**7**	**7**	**6**	**8**	**8**	**8**	**7**	**5**	**6**	**7**	**8**	**9**	**9**	**7**	**6**
	8	**8**	**8**	**8**	**9**	**9**	**8**	**7**	**8**	**9**	**7**	**6**	**5**	**6**	**5**	**5**	**9**	**11**	**10**	**10**
	PBM				**RT**				**SIL**				**WB**				**WG**			
	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20	0%	5%	10%	20%
S1	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
S2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	13	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	15	**15**	**15**	**15**	**15**	**15**	**15**	13	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
S3	4	4	4	4	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
	4	4	4	4	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
S4	5	5	4	4	**15**	**15**	**15**	**15**	**15**	14	**15**	14	**15**	**15**	**15**	**15**	17	16	17	**15**
	5	5	5	4	**15**	**15**	14	14	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
Sim2D2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	12	12	9	16	**2**	**2**	**2**	**2**
	2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	12	8	8	12	**2**	**2**	**2**	**2**
Sim5D2	**5**	**5**	**5**	4	3	3	3	3	3	3	3	3	**5**	**5**	**5**	4	3	3	3	3
	4	**5**	**5**	3	3	3	3	3	3	3	3	3	4	**5**	**5**	**5**	3	3	3	3
O200	3	3	4	4	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	19	19	20	18	**5**	**5**	20	20
	4	**5**	6	4	**5**	**5**	4	**5**	**5**	**5**	**5**	**5**	**5**	16	18	16	**5**	**5**	**5**	**5**
O2000	3	4	4	4	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
	4	4	4	4	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**	6	**5**	**5**	**5**	**5**
Ecoli	10	8	9	16	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	10	10	9	20	**3**	**3**	**3**	**3**
	3	**3**	**3**	2	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	17	**3**	**3**	**3**	**3**
Iris	**3**	**3**	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	17	20	19	20	**2**	**2**	**2**	**2**
	3	**3**	**3**	**3**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**	15	14	13	12	**2**	**2**	**2**	**2**
Seeds	**3**	**3**	10	**3**	2	2	2	2	2	2	2	2	**3**	**3**	19	20	2	2	2	2
	3	**3**	**3**	**3**	2	2	2	2	2	2	2	2	**3**	**3**	**3**	**3**	2	2	2	2
Total	**6**	**6**	**5**	**5**	**9**	**9**	**9**	**8**	**9**	**8**	**9**	**7**	**7**	**7**	**6**	**5**	**8**	**8**	**7**	**8**
	6	**8**	**7**	**5**	**9**	**9**	**7**	**7**	**9**	**9**	**9**	**9**	**8**	**8**	**8**	**6**	**9**	**9**	**9**	**9**

available data distance strategy. In the case of multidimensional data, we noticed that the quality of the clustering models highly depends on the form of the Inter term. In addition, significantly better validation results were achieved when the data sets resided in their original dimensions than in 2D presentation. The WG index clearly overperformed the other indices on the multidimensional sets.

The experiments to demonstrate the performance of the cluster validation indices were performed both on synthetic and real-world data sets. One challenge for testing indices with real-world data sets is that the correct number of clusters is not obvious. For instance, a clustering model may produce a useful presentation about the inherent structure of a data set while it does not necessarily agree with the given class distribution for the same data. In data mining, the goal of cluster analysis is, however, to discover new knowledge instead of training a prediction model in a supervised manner. In this scenario, one approach for validating a cluster model and estimating the number of clusters is to apply multiple indices that have previously performed well on several data sets.

We provided two modified versions of the original key point selection algorithm based on key point pruning. The developed algorithms included a mechanism for removing irrelevant key points. The algorithms resulted in good solutions for most of the data sets with a varying portion of missing values. However, there is still room for improvement in the heuristics to identify appropriate locations of the key points for diverse data sets. The development of clustering heuristics is not a trivial task because the notion of a cluster itself can be weakly defined [64]. It is also good to remember

TABLE 8. The number of clusters determined with internal cluster validation indices using K-spatialmedians clustering. The data sets consisted of predefined number of missing values, and different degrees of cluster overlap.

	CH				**DB**				**DB***				**GD**				**KCE**			
	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%
M10-dc0.9	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M10-dc0.8	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M10-dc0.7	4	4	4	4	8	10	10	10	7	10	10	10	**15**	**15**	**15**	**15**	2	2	2	2
M10-dc0.6	2	2	2	2	9	9	9	9	14	14	14	9	3	3	3	3	2	2	2	2
M50-dc0.9	3	3	3	3	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M50-dc0.8	2	2	2	2	13	13	13	13	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M50-dc0.7	3	3	3	3	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M50-dc0.6	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
M100-dc0.9	2	2	2	2	**15**	**15**	14	**15**	**15**	**15**	9	**15**	**15**	**15**	9	**15**	2	2	2	2
M100-dc0.8	2	2	2	2	**15**	**15**	14	**15**	**15**	**15**	14	**15**	**15**	**15**	12	**15**	2	2	2	2
M100-dc0.7	2	2	2	2	13	13	14	14	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	2	2	2	2
M100-dc0.6	3	3	3	3	13	11	10	11	6	3	3	3	**15**	**15**	3	**15**	2	2	2	2
Total	**0**	**0**	**0**	**0**	**6**	**6**	**4**	**6**	**8**	**8**	**6**	**8**	**10**	**10**	**7**	**10**	**0**	**0**	**0**	**0**

	PBM				**RT**				**SIL**				**WB**				**WG**			
	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20%	0%	5%	10%	20	0%	5%	10%	20%
M10-dc0.9	3	3	3	3	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	5	5	5	5	**15**	**15**	**15**	**15**
M10-dc0.8	3	2	2	2	**15**	**15**	**15**	**15**	8	8	8	**15**	7	7	7	8	**15**	**15**	**15**	**15**
M10-dc0.7	2	2	2	2	4	4	4	4	**15**	6	**15**	**15**	5	5	5	5	**15**	**15**	**15**	**15**
M10-dc0.6	2	2	2	2	9	9	9	13	3	3	3	3	3	3	3	3	**15**	**15**	**15**	**15**
M50-dc0.9	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	14	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**
M50-dc0.8	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	4	4	4	4	**15**	**15**	**15**	**15**
M50-dc0.7	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	14	12	5	5	5	5	**15**	**15**	**15**	**15**
M50-dc0.6	2	2	2	2	2	2	2	2	2	2	2	2	4	4	4	4	**15**	**15**	**15**	**15**
M100-dc0.9	2	2	2	2	**15**	**15**	14	**15**	**15**	**15**	16	**15**	5	5	5	5	**15**	**15**	14	**15**
M100-dc0.8	2	2	2	2	**15**	**15**	14	**15**	**15**	**15**	16	14	4	4	4	4	**15**	**15**	16	**15**
M100-dc0.7	2	2	2	2	**15**	**15**	**15**	**15**	**15**	**15**	**15**	**15**	3	3	4	3	**15**	**15**	**15**	**15**
M100-dc0.6	2	3	3	3	**15**	**15**	3	**15**	6	6	5	6	5	5	5	5	**15**	**15**	14	**15**
Total	**0**	**0**	**0**	**0**	**9**	**9**	**6**	**9**	**8**	**7**	**5**	**7**	**1**	**1**	**1**	**1**	**12**	**12**	**9**	**12**

that clustering is often in the eye of the beholder [65]. Before a clustering algorithm is applied to the data, one may also want to determine whether the data even has a clustering tendency [66]. The most central properties of clusters are density, variance, dimension, shape, and separation [67]. Further, what type of clustering model is the most useful always depends on the target application.

IX. CONCLUSION

Even though the basic idea behind cluster analysis is simple, the process presumes many decisions and choices with multiple options in different parts of the analysis. This study proposed a toolbox that enables researchers and practitioners to achieve reliable and consistent clustering results regardless of missing values in their data. The priorities of the present work were on data preprocessing, clustering, and cluster validation.

The toolbox supports missing values and enables its user to build automated data clustering pipelines from preprocessing to cluster analysis and model validation. The validity and performance of the algorithms were demonstrated using multiple test cases and several data sets. One should note that the aim of the presented experiments was not to perform a systematic method comparison since most of the underlying development work has already been accomplished in the previous studies cited in this paper.

We remind that some of the implemented functions can also be useful in other machine learning tasks. For instance, the distance computation methods for missing data cases provided in the preprocessing folder are readily applicable in supervised learning with the distance-based methods [68], [69].

The functionality of the toolbox was verified against the reference results from the previous publications. In the study, the two expected distances measuring metrics' performance were thoroughly demonstrated in handling missing values. Further, a recently published key point selection mechanism, which associates the data points with relatively higher density and larger density-based distances to the so-called key points, was applied to improve the cluster validation process. The cluster validation was experimented with challenging multidimensional data sets with various cluster overlap and numbers of missing values.

Even though the key point selection strategy seems to improve the performance of many cluster validation indices, further investigations are recommended, especially related to the key point selection procedure and the initialization of clustering algorithms. The initialization is an important part of the clustering process, and several studies are already available on the topic [33], [70]–[72]. The purpose of this toolbox is to facilitate and promote this research further. The UCI Repository provides a multitude of data sets, of which some are particularly proposed for clustering experiments. This toolbox enhances the testing of its methods with a wider range of sets.

APPENDIX A
EXPECTED SQUARED EUCLIDEAN DISTANCE

Let us assume the data are missing at random (MAR), i.e., missingness may depend on the value of available data:

$$P(M|\mathbf{x}_{avail}, \mathbf{x}_{miss}) = P(M|\mathbf{x}_{avail}). \tag{21}$$

The expected squared Euclidean distance between two data vectors can be partitioned into four parts depending on the missing and available values of each data vector:

$$\begin{aligned}
E\Big[||\mathbf{x}_i - \mathbf{x}_j||^2\Big]
&= \sum_{l\in A_i\cap A_j} ((\mathbf{x}_i)_l - (\mathbf{x}_j)_l)^2 + \sum_{l\in A_i\cap M_j} E[((\mathbf{x}_i)_l - (X_j)_l)^2] \\
&\quad + \sum_{l\in M_i\cap A_j} E[((X_i)_l - (\mathbf{x}_j)_l)^2] \\
&\quad + \sum_{l\in M_i\cap M_j} E[((X_i)_l - (X_j)_l)^2], \qquad (22)
\end{aligned}$$

where A_i and A_j denote the available values of data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, and M_i and M_j denote the missing values of the vectors. The first term ($l \in A_i \cap A_j$) represents pairwise known values of both vectors, and they can be computed directly. The rest of the sum contains terms where at least one part contains only missing values. The missing value can be replaced with a random value, i.e., $(\mathbf{x}_i)_l$ is denoted by $(X_i)_l$ for every $l \in M_i$. Thus, the equation can be expanded as follows:

$$\begin{aligned}
E\Big[||\mathbf{x}_i - \mathbf{x}_j||^2\Big]
&= \sum_{l\in A_i\cap A_j} \Big((\mathbf{x}_i)_l - (\mathbf{x}_j)_l\Big)^2 \\
&\quad + \sum_{l\in A_i\cap M_j} \Big(((\mathbf{x}_i)_l - E[(X_j)_l])^2 + Var[(X_j)_l]\Big) \\
&\quad + \sum_{l\in M_i\cap A_j} \Big((E[(X_i)_l] - (\mathbf{x}_j)_l)^2 + Var[(X_i)_l]\Big) \\
&\quad + \sum_{l\in M_i\cap M_j} \Big((E[(X_i)_l] - E[(X_j)_l])^2 \\
&\quad + Var[(X_i)_l] + Var[(X_j)_l]\Big). \qquad (23)
\end{aligned}$$

In more detail, the third summation ($l \in M_i \cap M_j$) can be written as:

$$\begin{aligned}
&E[((X_i)_l - (X_j)_l)^2] \\
&\quad = E[((X_i)_l)^2 - 2((X_i)_l)((X_j)_l) + ((X_j)_l)^2] \\
&\quad = E[((X_i)_l)^2] - 2E[((X_i)_l)]E[((X_j)_l)] + E[((X_j)_l)^2] \\
&\qquad + E[((X_i)_l)]^2 - E[((X_i)_l)]^2 + E[((X_j)_l)]^2 - E[((X_j)_l)]^2 \\
&\quad = (E[((X_i)_l)] - E[((X_j)_l)])^2 + E[E[(X_i)_l^2] - (X_i)_l^2] \\
&\qquad + E[E[(X_j)_l^2] - (X_j)_l^2] \\
&\quad = (E[((X_i)_l)] - E[((X_j)_l)])^2 + Var((X_i)_l) + Var((X_j)_l). \qquad (24)
\end{aligned}$$

Thus, it is sufficient to compute the expected value and variance of each random value separately to obtain the final distance.

APPENDIX B
CONDITIONAL MEAN AND COVARIANCE

Let us assume multivariate normally distributed data which are partitioned as $\mathbf{x} = \begin{bmatrix}\mathbf{x}_1 \\ \mathbf{x}_2\end{bmatrix}$ and define a linear combination $\mathbf{x} = \mathbf{x}_1 + \mathbf{A}\mathbf{x}_2$, where $\mathbf{A} = -\Sigma_{12}\Sigma_{22}^{-1}$. Now, we notice the following equality:

$$\begin{aligned}
Cov[\mathbf{x}, \mathbf{x}_2] &= Cov[\mathbf{x}_1, \mathbf{x}_2] + Cov[\mathbf{A}\mathbf{x}_2, \mathbf{x}_2] \\
&= \Sigma_{12} + \mathbf{A}Var[\mathbf{x}_2] \\
&= \Sigma_{12} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22} \\
&= \mathbf{0}.
\end{aligned}$$

Thus, $\mathbf{x}$ and $\mathbf{x}_2$ are uncorrelated. In addition, they are jointly normally distributed, and therefore, independent. Following the initial assumptions, the conditional mean of $\mathbf{x}_1$ given $\mathbf{x}_2$ is obtained as follows:

$$\begin{aligned}
E[\mathbf{x}_1|\mathbf{x}_2] &= E[\mathbf{x} - \mathbf{A}\mathbf{x}_2|\mathbf{x}_2] \\
&= E[\mathbf{x}|\mathbf{x}_2] - E[\mathbf{A}\mathbf{x}_2|\mathbf{x}_2] \\
&= E[\mathbf{x}] - \mathbf{A}\mathbf{x}_2 \\
&= \boldsymbol{\mu}_1 + \mathbf{A}(\boldsymbol{\mu}_2 - \mathbf{x}_2) \\
&= \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2).
\end{aligned}$$

Further, we find out the following equality:

$$\begin{aligned}
Var[\mathbf{x}_1|\mathbf{x}_2] &= Var[\mathbf{x} - \mathbf{A}\mathbf{x}_2|\mathbf{x}_2] \\
&= Var[\mathbf{x}|\mathbf{x}_2] + Var[-\mathbf{A}\mathbf{x}_2|\mathbf{x}_2] \\
&\quad + Cov[\mathbf{x}, -\mathbf{A}\mathbf{x}_2] + Cov[-\mathbf{A}\mathbf{x}_2, \mathbf{x}] \\
&= Var[\mathbf{x}|\mathbf{x}_2] + \mathbf{A}Var[\mathbf{x}_2|\mathbf{x}_2]\mathbf{A}^T \\
&\quad - Cov[\mathbf{x}, \mathbf{x}_2]\mathbf{A}^T - \mathbf{A}Cov[\mathbf{x}_2, \mathbf{x}] \\
&= Var[\mathbf{x}].
\end{aligned}$$

Therefore, the conditional variance is defined as:

$$\begin{aligned}
Var[\mathbf{x}_1|\mathbf{x}_2] &= Var[\mathbf{x}_1 + \mathbf{A}\mathbf{x}_2] \\
&= Var[\mathbf{x}_1] + \mathbf{A}Var[\mathbf{x}_2]\mathbf{A}^T \\
&\quad + Cov[\mathbf{x}_1, \mathbf{x}_2]\mathbf{A}^T + \mathbf{A}Cov[\mathbf{x}_2, \mathbf{x}_1] \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} \\
&\quad - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.
\end{aligned}$$

Note that the basic rules of matrix algebra are given in [73].

ACKNOWLEDGMENT

The authors acknowledge Joonas Hämäläinen, Ph.D., for his help in preparing the experiments reported in Table 8.

REFERENCES

[1] J. Kim, D. Tae, and J. Seok, "A survey of missing data imputation using generative adversarial networks," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 454–456.

[2] F. Biessmann, T. Rukat, P. Schmidt, P. Naidu, S. Schelter, A. Taptunov, D. Lange, and D. Salinas, "DataWig: Missing value imputation for tables," *J. Mach. Learn. Res.*, vol. 20, no. 175, pp. 1–6, 2019.

[3] J. K. Dixon, "Pattern recognition with partly missing data," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 10, pp. 617–621, Oct. 1979.

[4] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[5] E. Eirola, G. Doquire, M. Verleysen, and A. Lendasse, "Distance estimation in numerical data sets with missing values," *Inf. Sci.*, vol. 240, pp. 115–128, Aug. 2013.

[6] D. P. P. Mesquita, J. P. P. Gomes, A. H. S. Junior, and J. S. Nobre, "Euclidean distance estimation in incomplete datasets," *Neurocomputing*, vol. 248, pp. 11–18, Jul. 2017.

[7] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[8] S. Narayanan, R. J. Marks, J. L. Vian, J. J. Choi, M. A. El-Sharkawi, and B. B. Thompson, "Set constraint discovery: Missing sensor data restoration using autoassociative regression machines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2002, pp. 2872–2877.

[9] M. Abdella and T. Marwala, "The use of genetic algorithms and neural networks to approximate missing data in database," in *Proc. IEEE 3rd Int. Conf. Comput. Cybern. (ICCC)*, Apr. 2005, pp. 207–212.

[10] I. B. Aydilek and A. Arslan, "A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm," *Inf. Sci.*, vol. 233, pp. 25–35, Jun. 2013.

[11] M. G. Rahman and M. Z. Islam, "Missing value imputation using decision trees and decision forests by splitting and merging records: Two novel techniques," *Knowl.-Based Syst.*, vol. 53, pp. 51–65, Nov. 2013.

[12] L. Tran, X. Liu, J. Zhou, and R. Jin, "Missing modalities imputation via cascaded residual autoencoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1405–1414.

[13] N. Abiri, B. Linse, P. Edén, and M. Ohlsson, "Establishing strong imputation performance of a denoising autoencoder in a wide range of missing data problems," *Neurocomputing*, vol. 365, pp. 137–146, Nov. 2019.

[14] J. Zhao, Y. Nie, S. Ni, and X. Sun, "Traffic data imputation and prediction: An efficient realization of deep learning," *IEEE Access*, vol. 8, pp. 46713–46722, 2020.

[15] L. Li, M. Franklin, M. Girguis, F. Lurmann, J. Wu, N. Pavlovic, C. Breton, F. Gilliland, and R. Habre, "Spatiotemporal imputation of MAIAC AOD using deep learning with downscaling," *Remote Sens. Environ.*, vol. 237, Feb. 2020, Art. no. 111584.

[16] M. Sangeetha and M. S. Kumaran, "Deep learning-based data imputation on time-variant data using recurrent neural network," *Soft Comput.*, vol. 24, no. 17, pp. 13369–13380, Sep. 2020.

[17] S. Ryu, M. Kim, and H. Kim, "Denoising autoencoder-based missing value imputation for smart meters," *IEEE Access*, vol. 8, pp. 40656–40666, 2020.

[18] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: Incomplete multi-view clustering via contrastive prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11174–11183.

[19] G. Dong, G. Liao, H. Liu, and G. Kuang, "A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images," *IEEE Geosci. Remote Sens. Mag.*, vol. 6, no. 3, pp. 44–68, Sep. 2018.

[20] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 231–240, May/Jun. 2011.

[21] J. Tou and R. Gonzalez, *Pattern Recognition Principles*. Reading, MA, USA: Addison-Wesley, 1974.

[22] W. R. Dillon and M. Goldstein, *Multivariate Analysis: Methods and Applications*. Hoboken, NJ, USA: Wiley, 1984.

[23] M. R. Anderberg, *Cluster Analysis for Applications*. New York, NY, USA: Academic, 1973.

[24] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[25] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, 1988.

[26] C. K. Reddy and B. Vinzamuri, "A survey of partitional and hierarchical clustering algorithms," in *Data Clustering*. Boca Raton, FL, USA: CRC Press, 2018, pp. 87–110.

[27] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. Hoboken, NJ, USA: Wiley, 1990.

[28] J. Han, M. Kamber, and A. Tung, *Spatial Clustering Methods in Data Mining: A Survey*. New York, NY, USA: Taylor and Francis, 2001, pp. 188–217.

[29] J. Hämäläinen, S. Jauhiainen, and T. Kärkkäinen, "Comparison of internal clustering validation indices for prototype-based clustering," *Algorithms*, vol. 10, no. 3, p. 105, Sep. 2017.

[30] M. J. Zaki and W. Meira, *Data Mining and Analysis: Fundamental Concepts*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[31] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, and J. M. Pérez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognit.*, vol. 46, no. 1, pp. 243–256, Jan. 2013.

[32] Q. Li, S. Yue, and M. Ding, "Volume and surface area-based cluster validity index," *IEEE Access*, vol. 8, pp. 24170–24181, 2020.

[33] J. Hämäläinen, T. Kärkkäinen, and T. Rossi, "Improving scalable K-means++," *Algorithms*, vol. 14, no. 1, p. 6, Dec. 2020.

[34] T. Kärkkäinen and J. Toivanen, "Building blocks for odd–even multigrid with applications to reduced systems," *J. Comput. Appl. Math.*, vol. 131, nos. 1–2, pp. 15–33, Jun. 2001.

[35] M. Saarela and T. Karkkainen, "Discovering gender-specific knowledge from Finnish basic education using Pisa scale indices," in *Proc. 7th Int. Conf. Educ. Data Mining*, 2014, pp. 60–67.

[36] M. Niemela, S. Ayramo, and T. Karkkainen, "Comparison of cluster validation indices with missing data," in *Proc. 26th Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, 2018, pp. 461–466.

[37] H. Fischer, "A history of the central limit theorem," in *Sources and Studies in the History of Mathematics and Physical Sciences*. New York, NY, USA: Springer, 2011.

[38] T. Karkkainen and M. Saarela, "Robust principal component analysis of data with missing values," in *Machine Learning and Data Mining in Pattern Recognition*, P. Perner, Ed. Cham, Switzerland: Springer, 2015, pp. 140–154.

[39] E. Eirola, A. Lendasse, V. Vandewalle, and C. Biernacki, "Mixture of Gaussians for distance estimation with missing data," *Neurocomputing*, vol. 131, pp. 32–42, May 2014.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Roy. Stat. Soc., Ser. B, Methodol.*, vol. 39, no. 1, pp. 1–22, 1977.

[41] F. Dellaert, "The expectation maximization algorithm," in *College of Computing*. Atlanta, GA, USA: Georgia Institute of Technology, 2003.

[42] J. Van Hulse and T. M. Khoshgoftaar, "Incomplete-case nearest neighbor imputation in software measurement data," *Inf. Sci.*, vol. 259, pp. 596–610, Feb. 2014.

[43] A. Majumdar and R. K. Ward, "Some empirical advances in matrix completion," *Signal Process.*, vol. 91, no. 5, pp. 1334–1338, 2011.

[44] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 2287–2322, Aug. 2019.

[45] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[46] A. Mead, "Review of the development of multidimensional scaling methods," *J. Roy. Stat. Soc., D Statistician*, vol. 41, no. 1, pp. 27–39, 1992.

[47] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–136, Mar. 1982.

[48] M. E. Celebi, H. A. Kingravi, and P. A. Vela, "A comparative study of efficient initialization methods for the k-means clustering algorithm," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 200–210, Jan. 2013.

[49] S. Ayramo and T. Karkkainen, "Introduction to partitioning-based clustering methods with a robust example," in *Reports of the Department of Mathematical Information Technology Series C. Software and Computational Engineering*. Jyväskylä, Finland: Univ. Jyvaskyla, 2006.

[50] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, Jun. 2005.

[51] S. Ayramo, *Knowledge Mining Using Robust Clustering. Jyvaskyla Studies in Computing*, vol. 63. Jyväskylä, Finland: Univ. Jyvaskyla, 2006.

[52] T. Karkkainen and E. Heikkola, "Robust formulations for training multilayer perceptrons," *Neural Comput.*, vol. 16, pp. 837–862, Apr. 2004.

[53] T. Karkkainen and S. Ayramo, "On computation of spatial median for robust data mining," in *Evolutionary and Deterministic Methods for Design Optimization and Control With Applications to Industrial and Societal Problems*. Munich, Germany: EU-ROGEN, 2005.

[54] E. Lughofer, "A dynamic split-and-merge approach for evolving cluster models," *Evolving Syst.*, vol. 3, no. 3, pp. 135–151, Sep. 2012.

[55] S. Shatskikh and L. E. Melkumova, "Normality assumption in statistical data analysis," in *Proc. CEUR Workshop*, vol. 1638, 2016, pp. 763–768.

[56] M. Niemelä and T. Kärkkäinen, "Improving clustering and cluster validation with missing data using distance estimation methods," in *Computational Sciences and Artificial Intelligence in Industry: New Digital Technologies for Solving Future Societal and Economical Challenges*, T. Tuovinen, J. Periaux, and P. Neittaanmäki, Eds. Springer, 2022, pp. 123–133.

[57] M. C. F. D. Oliveira and H. Levkowitz, "From visual data exploration to visual data mining: A survey," *IEEE Trans. Vis. Comput. Graphics*, vol. 9, no. 3, pp. 378–394, Jul. 2003.

[58] C. Boutsidis, A. Zouzias, and P. Drineas, "Random projections for k-means clustering," in *Proc. Adv. Neural Inf. Process. Syst., 24th Annu. Conf. Neural Inf. Process. Syst.*, 2010, pp. 298–306.

[59] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Cham, Switzerland: Springer, 2002.

[60] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. Patel, A. Tiwari, M. Er, W. Ding, and C. T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, Dec. 2017.

[61] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

[62] D. Dua and C. Graff. UCI Machine Learning Repository. School of Information and Computer Sciences. University of California, Irvine. Accessed: 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[63] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, vol. 3512, 2005, pp. 758–770.

[64] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Reading, MA, USA: Addison-Wesley, 2005.

[65] V. Estivill-Castro, "Why so many clustering algorithms: A position paper," *ACM SIGKDD Explor. Newslett.*, vol. 4, no. 1, pp. 65–75, Jun. 2002.

[66] S. P. Smith and A. K. Jain, "Testing for uniformity in multidimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 1, pp. 73–81, Jan. 1984.

[67] M. S. Aldenderfer and R. K. Blashfield, *Cluster Analysis*. Newbury Park, CA, USA: Sage, 1984.

[68] T. Kärkkäinen, "Extreme minimal learning machine: Ridge regression with distance-based basis," *Neurocomputing*, vol. 342, pp. 33–48, May 2019.

[69] J. Hämäläinen, A. S. C. Alencar, T. Kärkkäinen, C. L. C. Mattos, A. H. S. Júnior, and J. P. P. Gomes, "Minimal learning machine: Theoretical results and clustering-based reference point selection," *J. Mach. Learn. Res.*, vol. 21, pp. 1–29, Oct. 2020.

[70] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," *Pattern Recognit. Lett.*, vol. 25, no. 11, pp. 1293–1302, 2004.

[71] M. Marina and H. David, "An experimental comparison of several clustering and initialization methods," in *Proc. 14th Annu. Conf. Uncertainty Artif. Intell. (UAI)*. San Francisco, CA, USA: Morgan Kaufmann, 1998, pp. 386–395.

[72] S. Ayramo, T. Karkkainen, and K. Majava, "Robust refinement of initial prototypes for partitioning-based clustering algorithms," in *Recent Advances in Stochastic Modeling and Data Analysis*. Singapore: World Scientific, 2007, pp. 473–482.

[73] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook. Technical University of Denmark. Accessed: 2012. [Online]. Available: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

MARKO NIEMELÄ received the M.Sc. (technology) degree in computer science and engineering from the University of Oulu, in 2013. He is currently working with the Faculty of Information Technology, University of Jyväskylä, Finland. His main research interests include machine learning, data mining, data analytics, and optimization.

SAMI ÄYRÄMÖ received the Ph.D. degree in mathematical information technology, in 2006. He is currently an Adjunct Professor of data analytics at the University of Jyväskylä. His research interests include machine learning and predictive modeling with a special focus on applications in sport, health, and medicine.

TOMMI KÄRKKÄINEN (Senior Member, IEEE) received the Ph.D. degree in mathematical information technology from the University of Jyväskylä (JYU), in 1995. Since 2002, he has been a Full Professor of mathematical information technology with the Faculty of Information Technology (FIT), JYU. He currently leads the Research Division and the Research Group on human and machine-based intelligence in learning. He has served in many administrative positions at FIT and JYU. He has led nearly 50 different research and development projects. He has been/is involved in supervising 57 Ph.D. students. He has published about 200 peer-reviewed articles. His current research interests include data mining, machine learning, learning analytics, and computing education research. He received the JYU Innovation Prize, in 2010.