MOHAMMAD REZAEI

Clustering validation

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 225

Academic Dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in Louhela auditorium in Science Park Building at the University of Eastern Finland, Joensuu, on June 10, 2016, at 12 o'clock noon.

School of Computing

Grano Oy
Joensuu, 2016
Editor: Dr. Pertti Pasanen

Distribution:
University of Eastern Finland Library / Sales of publications
P.O.Box 107, FI-80101 Joensuu, Finland
tel. +358-50-3058396
http://www.uef.fi/kirjasto

ISBN: 978-952-61-2144-4 (printed)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-2145-1 (pdf)
ISSNL: 1798-5668
ISSN: 1798-5676

Author: Mohammad Rezaei
University of Eastern Finland
School of Computing
P.O.Box 111
80101 JOENSUU
FINLAND
email: rezaei@cs.uef.fi

Supervisor: Professor Pasi Fränti, PhD.
University of Eastern Finland
School of Computing
P.O.Box 111
80101 JOENSUU
FINLAND
email: franti@cs.uef.fi

Reviewers: Professor Ana Luisa N. Fred, PhD
Instituto Superior Técnico
Torre Norte, Instituto de Telecomunicações
Av. Rovisco Pais, 1
1049-001, Lisbon
PORTUGAL
email: afred@lx.ir.pt

Professor James Bailey, PhD
University of Melbourne
Department of Computing and Information Systems
Victoria 3010
AUSTRALIA
email: baileyj@unimelb.edu.au

Opponent: Professor Ioan Tabus, PhD
Tampere University of Technology
Department of Signal Processing
P.O.Box 527
33101 Tampere
FINLAND
email: ioan.tabus@tut.fi

ABSTRACT:

Cluster analysis or clustering is one of the most fundamental and essential data mining tasks with broad applications. It aims at finding a structure in a set of unlabeled data, producing clusters so that objects in one cluster are similar in some way and different from objects in other clusters. Basic elements of clustering include proximity measure between objects, cost function, algorithm, and cluster validation. There is a close relationship between these elements. Although there has been extensive research on clustering methods and their applications, less attention has been paid to the relationships between the basic elements. This thesis first provides an overview of the basic elements of cluster analysis. It then focuses on cluster validity as four publications are devoted to this element.

Chapter 1 sketches the clustering procedure and provides definitions of basic components. Chapter 2 reviews popular proximity measures for different types of data. A novel similarity measure for comparing two groups of words is introduced which is used in the clustering of items characterized by a set of keywords. Chapter 3 presents basic clustering algorithms and Chapter 4 analyzes cost functions. A clustering algorithm is expected to optimize a given cost function. However, in many cases the cost function is unknown and hidden with the algorithm, making the evaluation of clustering results and analysis of the algorithms difficult.

Numerous clustering algorithms have been developed for different application fields. Different algorithms, or even one algorithm with different parameters, can give different results for the same data set. The best clustering can be selected based on the cost function if the number of clusters is fixed and the cost function has been defined, otherwise cluster validity indices, internal and external, are used. Chapter 5 reviews several popular internal indices. We study the problem of determining the number of clusters in a data set using these indices, and we propose a new internal index for finding the number of clusters in hierarchical clustering of words. External

validity indices are studied in Chapter 6 and two new external indices, centroid index and pair sets index, are introduced. We present a novel experimental setup based on generated partitions to evaluate external indices. We also study whether external indices are applicable to the problem of determining the number of clusters. The conclusion is made that external indices can be used for the problem, but only in theory and in controlled environments where the type of data is well known and no surprises appear. In practice, this is rarely the case.

AMS classification: 62H30, 91C20

Universal Decimal Classification: 004.052.42, 303.722.4, 519.237.8

Library of Congress Subject Headings: Data mining; cluster analysis; algorithms

Yleinen suomalainen asiasanasto: tiedonlouhinta; klusterianalyysi; validointi; algoritmit

Preface

This PhD dissertation contains the results of research completed at the School of Computing of the University of Eastern Finland during the years 2012-2016. Many individuals have helped me both directly and indirectly in my research and writing this thesis.

I would like to express my sincere gratitude to my supervisor, Professor Pasi Fränti, for giving me the chance to study in the PhD program and for his support with research throughout the years. I would never have finished this dissertation without his help and guidance.

I would also like to thank my colleagues who helped me during my PhD study specially Dr. Qinpei Zhao.

I am thankful to Professor Ana Luisa N. Fred and Professor James Bailey, the reviewers of the thesis, for their feedback and comments.

I extend my heartfelt gratitude to my father and mother, my first teachers. Thank you so much for your help and support. I would like to express my deepest love and gratitude to my wife and my sons.

This research has been supported by MOPSI and MOPIS projects, SCITECO and LUMET grants from University of Eastern Finland, and the Nokia FOUNDATION.

Joensuu, May 9, 2016 *Mohammad Rezaei*

LIST OF ORIGINAL PUBLICATIONS

P1 P. Fränti, M. Rezaei, Q. Zhao, "Centroid index: cluster level similarity measure", *Pattern Recognition*, 47(9), pp. 3034-3045, 2014.

P2 M. Rezaei, P. Fränti, "Set matching measures for external cluster validity", *IEEE Transactions on Knowledge and Data Engineering*, 2016, (accepted).

P3 M. Rezaei, P. Fränti, "Can number of clusters be solved by external validity index?", 2016, (submitted).

P4 Q. Zhao, M. Rezaei, P. Fränti, "Keyword clustering for automatic categorization", *International Conference on Pattern Recognition* (*ICPR*), pp. 2845-2848, 2012.

P5 M. Rezaei, P. Fränti, "Matching similarity for keyword-based clustering", *Joint IAPR International Workshop*, *SSPR & SPR* 2014, Joensuu, (S+SSPR), pp. 193-202, 2014.

Throughout the thesis, these papers will be referred to by [P1]-[P5]. These papers are included at the end of this thesis by the permission of their copyright holders.

AUTHOR'S CONTRIBUTION

The idea of the paper [P1] originates from Prof. Pasi Fränti. The author contributed by refining the definition of the centroid index and extending it to the corresponding point-level index. The principal ideas of the other papers originate from the author. Implementations for the papers [P2], [P3], and [P5] were performed completely by the author. The author implemented the point-level index in [P1]. Implementation of the idea in [P4] was done by the author, except the libraries and similarity measures using WordNet.

The author performed all experiments for [P2]-[P5] and part of the experiments for [P1].

[P1] was written by Prof. Pasi Fränti and [P4] by Dr. Qinpei Zhao. The author helped to refine the text and provided materials for some sections of the papers. The author has written the papers [P2], [P3], and [P5].

List of symbols

N	number of data objects
X	data object as vector
x_i	i^{th} data objects
P_i	i^{th} cluster of clustering solution P
K	number of clusters
c_i	centroid of the i^{th} cluster
n_i	number of objects in the i^{th} cluster
$\bar{x}$	average of all data objects
D	dimension of data

Contents

1 Introduction

Clustering is the division of data objects into groups or clusters such that objects in the same group are more similar than objects in different groups. Clustering plays an important role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, customer relationship management (CRM), marketing, medical diagnostics, computational biology, and visualization [1].

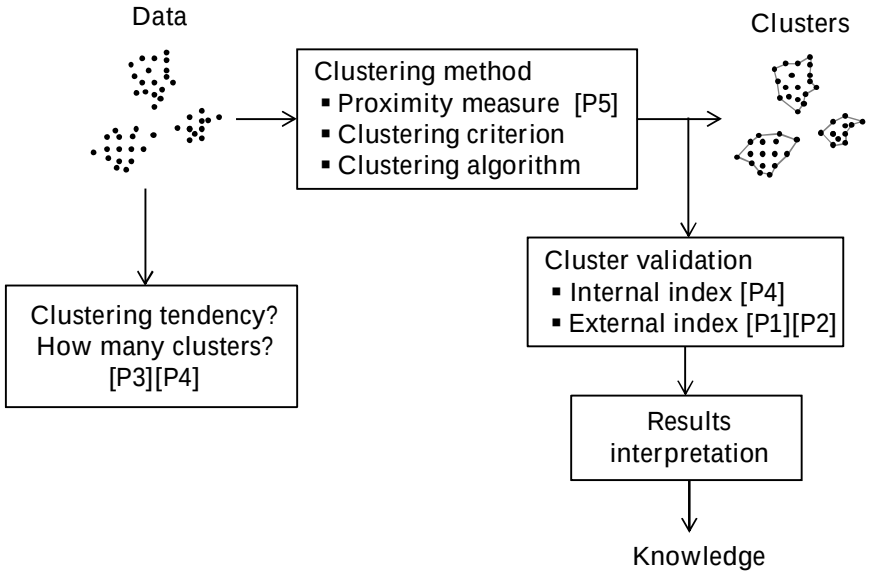


Figure 1.1: Basic components of cluster analysis

Figure 1.1 shows the components of cluster analysis. Data is represented in terms of *features* that form *d*-dimensional feature vectors. *Feature extraction* and selection from original entities must be performed so that the features provide as much distinction as possible between different entities concerning the task of interest. This is performed by an expert in the field. For example, the extraction of features from a speech signal to

distinguish between different people is performed by an expert in the speech processing field [2]. Moreover, extracted features may need preprocessing, such as dimensionality reduction and normalization of the features, so that all features have the same scale and contribute equally. Next, the assumption is made that the features have been already extracted and the required preprocessing has been performed. The basic components of cluster analysis are the following:

1. Proximity measure
2. Clustering criterion
3. Clustering algorithm
4. Cluster validation
5. Results interpretation

Similarity or *dissimilarity* (*distance*) measure between two data objects is a basic requirement for clustering, and it is chosen based on the problem at hand. For example, suppose that the problem concerns a time analysis of travelling in a city. Using Euclidean distance between two places is not accurate because one cannot typically travel through buildings. We study several proximity measures in Chapter 2 including a new similarity between two groups of words.

Clustering criterion determines the type of clusters that are expected. The criterion is expressed as a *cost* (or *objective*) *function*, or some other rules. For example, for the same data set, one criterion leads to hyperspherical clusters, whereas another leads to elongated clusters [2]. The cost function is hidden in many existing clustering approaches, however, the function can be determined through further analysis. We study several cost functions in Chapter 4.

Clustering algorithm is the procedure that groups data in order to optimize the clustering criterion. Numerous clustering algorithms have been developed for different fields. Good algorithms find a clustering close to the optimum efficiently. In Chapter 3, we review basic clustering algorithms.

Different clustering algorithms, and even one algorithm with different parameters and initial assumptions, can produce different clusterings for the same data set. For a fixed number of

clusters, different results can be evaluated based on the clustering criterion if available. In a general case, *cluster validation* techniques are used to evaluate the results of a clustering algorithm [3], and decide which clustering best fits the data. Cluster validation is performed using cluster validity indices which are divided into two groups: *internal index* and *external index* [P2].

Internal indices measure the quality of a clustering solution using only the underlying data [4], [5]. External indices compare two clustering solutions of the same dataset. They might compare a clustering with ground truth to evaluate a clustering algorithm. Both internal and external indices are used for determining the number of clusters. We study cluster validity indices in Chapters 5 and 6.

The goal of clustering is to provide meaningful insights to the data in order to develop a better understanding of the data. Therefore, in many cases, the expert in the application field is encouraged to interpret the resulting partitions and integrate the results with other experimental evidence and analysis in order to draw the right conclusions.

2 Proximity measures

A data object represents an entity and is described by attributes or features with a certain type, such as a number or a word. Attributes are often represented by a multidimensional vector [6]. The type of attributes is one of the factors that determines how to measure the similarity between two objects. Other factors are related to the problem at hand. For example, the similarity of two words for some applications is measured by considering the letters in the words. However, for other applications, this does not provide good results, and the semantic similarity between two words is required.

A dissimilarity or similarity measure can be effective without being a metric [7], but sometimes metric requirements are desirable. A dissimilarity *metric* must satisfy the following conditions [7]:

Non-negativity:	$D(x_i, x_j) \geq 0$
Symmetry:	$D(x_i, x_j) = D(x_j, x_i)$
Reflexivity:	$D(x_i, x_j) = 0$ if and only if $x_i=x_j$.
Triangular inequality:	$D(x_i, x_j)+ D(x_j, x_k)\geq D(x_i, x_k)$

A similarity metric satisfies the following:

Limited range:	$S(x_i, x_j) \leq S_0$
Symmetry:	$S(x_i, x_j) = S(x_j, x_i)$
Reflexivity:	$S(x_i, x_j) = S_0$ if and only if $x_i=x_j$.
Triangular inequality:	$S(x_i, x_j)\times S(x_j, x_k) \leq S(x_i, x_k)\times(S(x_i, x_j)+S(x_j, x_k))$

2.1 ELEMENTARY DATA TYPES

Numeric: Numeric data are classified in two groups: interval and ratio. The interval between each consecutive point of measurement is equal to every other for *interval* data, such as

time and temperature. They do not have a meaningful zero point. For example, 00.00 am is not the absence of time. The difference between 10:15 and 10:30 has exactly the same value as the difference between 8:00 and 8:15. In *ratio* data, such as the number of people in line, a value of zero indicates an absence of whatever is measured. Another classification for numeric data includes discrete data and continuous data.

Categorical: Every object belongs to one of a limited number of possible categories, states, or names. Categorical data are classified into two groups: nominal and ordinal. Categories in *nominal* data such as marriage status (married, widow, single) are not ordered. Binary data can be considered as nominal data with only two states: 0 and 1. On the other hand, categories in *ordinal* data, such as degree of pain (severe, moderate, mild, none) are ordered.

2.2 NUMERICAL DISTANCES

Euclidean distance

Euclidean distance is the most common metric that is used for numerical vector objects. For two d dimensional objects x_i and x_j, Euclidean distance is calculated as follows:

$$d = \left(\sum_{l=1}^{d} \left| x_i^l - x_j^l \right|^2 \right)^{1/2} \tag{2.1}$$

Centroid-based clustering algorithms, such as K-means, that use Euclidean distance tend to provide hyperspherical clusters [6].

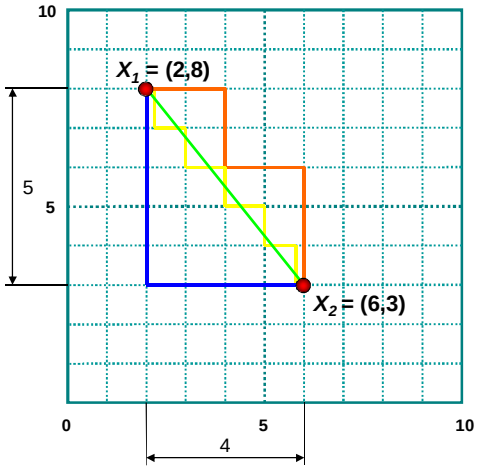
Euclidean distance is a special case (p=2) of a more general metric called Minkowski distance:

$$d = \left(\sum_{l=1}^{d} \left| x_i^l - x_j^l \right|^p \right)^{1/p} \tag{2.2}$$

Another popular and special case of Minkowski distance is Manhattan or city-block distance where p=1, see Figure 2.1:

$$d = \sum_{l=1}^{d} \left| x_i^l - x_j^l \right| \tag{2.3}$$

A clustering algorithm that uses Manhattan distance tends to build hyper-rectangular clusters [6].



2D example

x_1 = (2,8)

x_2 = (6,3)

Euclidean distance

$$d(1,2) = \sqrt{|2-6|^2 + |8-3|^2} = \sqrt{41}$$

Manhattan distance

$$d(1,2) = |2-6| + |8-3| = 9$$

Figure 2.1: Euclidean and Manhattan distances (http://cs.uef.fi/pages/franti/cluster/notes.html)

Mahalonobis distance

All the objects in a cluster affect on Mahalonobis distance between two objects by applying within group covariance matrix *S*. Clustering algorithms that use this distance tend to build hyper-ellipsoidal clusters.

$$d = (x_i - x_j)^T S^{-1} (x_i - x_j) \tag{2.4}$$

The within group covariance matrix for uncorrelated features becomes an identity matrix and, therefore, Mahalonobis distance simplifies to Euclidean distance [6].

2.3 NON-NUMERICAL DISTANCES

Cosine similarity

Cosine similarity is the most popular metric used in document clustering and is based on the angle between the vectors of two objects.

$$s = \frac{X_i \bullet X_j}{\|X_i\|\|X_j\|} \tag{2.5}$$

The more similar two objects are, the more parallel they are in the feature space, and the greater the cosine value. The Cosine value does not provide information on the magnitude of the difference.

Hamming distance

Hamming distance is used for comparing categorical data and strings of equal length. It counts the number of different elements in two objects [8]:

$$d = \sum_{l=1}^{d} d_l(x_i^l, x_j^l), \qquad d_l(x_i^l, x_j^l) = \begin{cases} 0, & x_i^l = x_j^l \\ 1, & x_i^l \neq x_j^l \end{cases} \tag{2.6}$$

Following are some examples:

Cables, Tablet	*d*=2
10110001, 11100101	*d*=3
(male, blond, blue, A), (female, blond, brown, A)	*d*=2

Gower similarity is a variant of Hamming distance, which is normalized by the number of attributes and has been extended for mixed categorical and numerical data [9]. The simple form of Gower similarity for categorical data can be written as follows:

$$S = \frac{\sum_{l=1}^{d} S_l(x_i^l, x_j^l)}{d}, \; S_l(x_i^l, x_j^l) = \begin{cases} 1, & x_i^l = x_j^l \\ 0, & x_i^l \neq x_j^l \end{cases} \tag{2.7}$$

Edit distance

Levenshtein or *edit distance* measures the dissimilarity of two strings (e.g., words) by counting the minimum number of insertions, deletions, and substitutions required to transform one string to the other. Several variants exist. For example, *longest common subsequence* (LCS) allows only insertions and deletions [10]. We describe the edit distance by an example: the dissimilarity between *kitten* and *sitting*. Transforming *kitten* into *sitting* can be performed in three steps as follows:

Substitute *s* with *k*: sitten
Substitute *e* with *i*: sittin
Insert *g* at the end: sitting

Therefore, the edit distance between the two words is 3.

2.4 SEMANTIC SIMILARITY BETWEEN WORDS

Semantic similarity between two words is measured according to their meaning rather than their syntactical representation. Measures for the semantic similarity of words can be categorized as *corpus-based*, *search engine-based, knowledge-based* and *hybrid*. Corpus-based measures such as *point-wise mutual information* (PMI) [11] and *latent semantic analysis* (LSA) [11] define the similarity based on large corpora and term co-occurrence. The number of occurrences and co-occurrences of two words in a large number of documents is used to approximate their similarity. A high similarity is achieved when the number of co-occurrences is only slightly lower than the number of occurrences of each word. Search engine-based measures such as *Google distance* are based on web counts and snippets from the results of a search engine [12] [13] [14]. *Flickr distance* first searches for two target words separately through image tags and then uses image content to calculate the distance between two words [15].

Knowledge-based measures use lexical databases such as *WordNet* [16] or *CYC* [16]. These databases can be considered computational formats of large amounts of human knowledge. The knowledge extraction process is time consuming and the database depends on human judgment. Moreover, it does not scale easily to new words, fields, and languages [17] [18].

WordNet is a taxonomy that requires a procedure to derive a similarity score between words. Despite its limitations, it has been successively used for clustering [P4]. Figure 2.2 illustrates a small part of the WordNet hierarchy where mammal is the *least subsummer* of wolf and hunting dog. *Depth* of a word is the number of links between it and the root word in WordNet. As an example, the Wu and Palmer measure [19] is defined as follows:

$$S(w_1, w_2) = \frac{2 \times depth(LCS(w_1, w_2))}{depth(w_1) + depth(w_2)} \tag{2.8}$$

where *LCS* is the least common subsummer of the words w_1 and w_2.

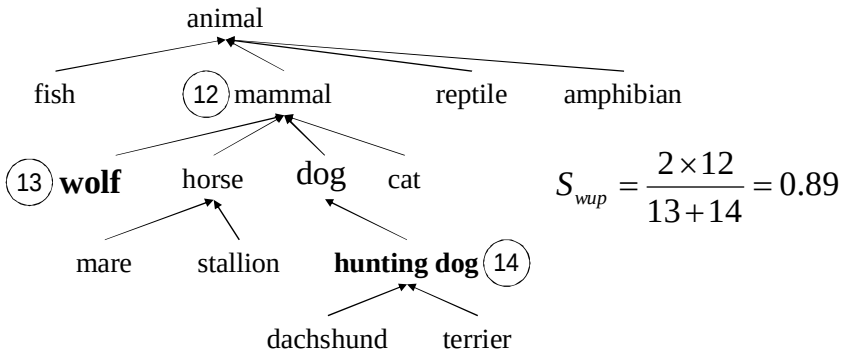


Figure 2.2: Part of WordNet taxonomy

Jiang-Contrath [16] is a hybrid of corpus-based and knowledge-based methods in that it extracts the information content of two words and their least subsumer in a corpus. Methods based on Wikipedia or similar websites are also hybrid in the sense that they use organized corpora with links between documents [20].

2.5 SEMANTIC SIMILARITY BETWEEN GROUPS OF WORDS

The semantic clustering of objects such as documents, web sites, and movies based on their keywords requires a similarity measure between two sets of keywords. Existing measures include minimum, maximum, and average similarity. Consider the bipartite graph in Figure 2.3 where the similarity between every two words is written on their corresponding link. Minimum and maximum measures are based on the links with minimum (0.20) and maximum (0.84) values. The average measure considers all the links and calculates the average value (0.57). These measures have fundamental limitations in providing a reasonable similarity value between two sets of words [P5]. For example, the minimum and average measures give a lower value than 1.00 for two sets with the same words. Maximum measure gives 1.00 for two different sets which have only one common word.

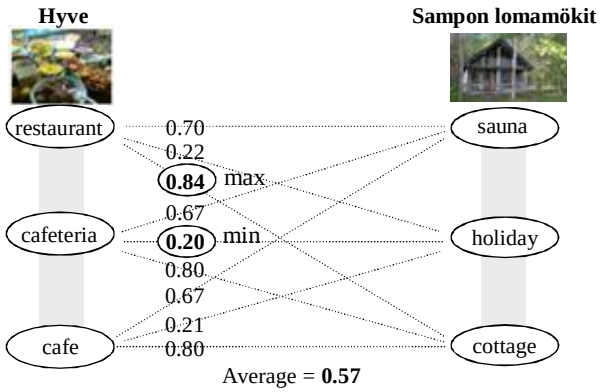


Figure 2.3: Minimum and maximum similarities between two location-based services is derived by considering two keywords with minimum and maximum similarities

In [P5], we present a new measure based on matching the words of two groups assuming that a similarity measure between two individual words is available. The proposed *matching similarity* measure is based on a greedy pairing algorithm which first finds the two most similar words across

the sets, and then iteratively matches next similar words. Finally, the remaining non-paired keywords (of the object with more keywords) are just matched with the most similar words in the other object. Figure 2.4 illustrates the matching process between two sample objects.

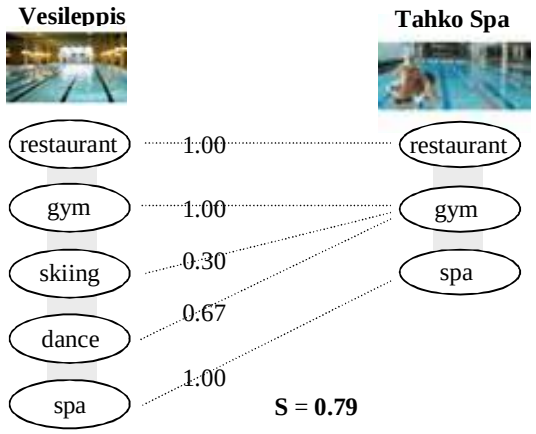


Figure 2.4: Matching between the words of two objects.

Consider two objects with N_1 and N_2 keywords so that $N_1 > N_2$. We define normalized similarity between the two objects as follows:

$$S = \frac{\sum_{i=1}^{N_1} S(w_i, w_{p(i)})}{N_1} \tag{2.9}$$

where $S(w_i,w_j)$ measures the similarity between two words, and $p(i)$ provides the matched word for w_i in the other object. The proposed measure eliminates the disadvantages of minimum, maximum, and average similarity measures.

3 Clustering algorithms

3.1 K-MEANS

K-means is a partitional clustering algorithm that aims at minimizing the total squared error (TSE). To cluster N data objects into K clusters, K centroids are initially selected in some way, for example, through randomly chosen data objects. Two steps of the algorithm are then iteratively performed: *assignment* and *update*, for a fixed number of iterations or until convergence. In the first step, objects are assigned to their nearest centroid. In the second step, new centroids are calculated by averaging the objects in each cluster [21]. Time complexity is $O(IKN)$, where I is the number of iterations [22].

K-means suffers from several drawbacks [6]. The main drawback is that the result is highly dependent on the initial selection of centroids. Different centroids lead to different local optimums that may be very far away from the global one. Consequently, many variants of K-means have been proposed to tackle the obstacles. For example, several techniques such as K-means++ [23] have been proposed for the better selection of initial centroids. Iterative methods such as genetic algorithm [24] and random swap [25] improve results by modifying the centroids.

3.2 RANDOM SWAP

The *randomized local search* or *random swap* algorithm [25] selects one of the centroids in a given clustering randomly and moves it to another location. K-means is then applied to fine tune the clustering result. The process is repeated for a given number of iterations chosen as an input parameter. In each iteration, the new resulting clustering is accepted if it improves TSE, and is

then used for the next iteration. With large number of iterations, typically 5,000, the method usually provides good results. This trial-and-error approach is simple to implement and very effective in practice.

3.3 AGGLOMERATIVE CLUSTERING

Agglomerative clustering is a bottom-up approach in which each object is initially considered as its own cluster. Two clusters are then iteratively merged based on a criterion [26]. Several criteria have been proposed for selecting the next two clusters to be merged such as *single-linkage*, *average-linkage*, *complete-linkage*, *centroid-linkage*, and *Ward's method* [27].

Classical agglomerative clustering using any of these criteria is not appropriate for large-scale data sets due to the quadratic computational complexities in both execution time and storing space. The time complexity of the basic agglomerative clustering is $O(N^3)$. The fast algorithm introduced in [28] employs a nearest neighbor table that only uses $O(N)$ memory and reduces the time complexity to $O(\alpha N^2)$, where $\alpha << N$. Even this algorithm can still be too slow for real-time applications. In [26], an algorithm based on k-nearest neighbor graph is proposed to improve the speed close to $O(N \log N)$ with a slight decrease in accuracy. However, graph creation is the bottleneck of the algorithm and should be solved. Otherwise, this step dominates the time complexity. Agglomerative clustering is sensitive to noise and outliers. It does not consider an object after it is assigned to a cluster, and therefore, previous misclassifications cannot be corrected afterwards [6].

3.4 DBSCAN

Density Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm which aims at finding arbitrary shaped clusters and eliminate noise. It

creates clusters from the points whose neighborhood within a given radius (*eps*) contains a minimum number (*minPt*) of other points [29]. Using every such a point, the algorithm grows a cluster by joining other points that are close to the cluster. The results are independent of the order of processing the objects.

Three types of points are defined, see Figure 3.1. *Core* points contain at least *minPt* (5 in this example) points in their *eps* neighborhood. *Border* points do not contain enough points in their neighborhood but they fall in the neighborhood of some core points. Other points are considered *noise* or *outliers*.

A point x_i is directly density reachable from x_j if x_j is a core point and x_i is in its eps neighborhood. A point x_i is defined density reachable from a core point x_j if a chain of points from x_j to x_i exist so that each point is directly density reachable from the previous point. The concept of density connectivity is also defined to describe the relations between the border points that belong to the same cluster but are not density reachable from each other. Two points are density connected if they are density reachable from a common core point. A cluster is built from a core point and its neighboring objects in eps distance, and it grows using the concepts of density-reachable and density-connected. Two conditions should be held:

1. If x_i is in cluster *C*, and x_j is density reachable from x_i, then x_j also belongs to cluster *C*
2. If x_i and x_j belongs to cluster *C*, x_i and x_j are density connected

The results are highly dependent on the input parameters *eps* and *minPt*. Finding appropriate parameters for a data set is not trivial, and the problem becomes more complicated when different parts of data require different parameters [1]. Several methods such as Ordering Points To Identify the Clustering Structure (OPTICS) [30] have been proposed to address this problem. Time complexity of the original DBSCAN is $O(N^2)$ but efforts [31] [32] have been made to reduce it close to $O(N)$.

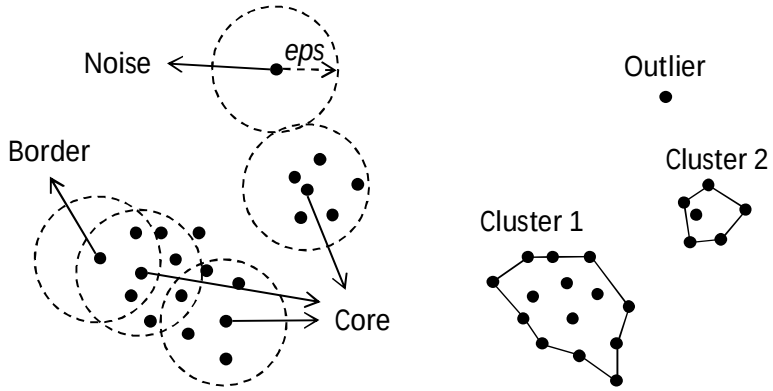


Figure 3.1: Three types of points are defined in the DBSCAN algorithm; two clusters are identified in this example, where *eps*=1 and *minPt*=5.

4 Cost functions

An objective function or cost function measures the error in a clustering. The optimal clustering is achieved by minimizing the cost function. However, not all clustering algorithms are based on minimizing a cost function. Some include the cost function hidden within the algorithm. This makes the evaluation of clustering results and analysis of the algorithms difficult. For example, DBSCAN produces a clustering heuristically with two given input parameters. Different parameter values result in different clusterings. No objective function has been reported to decide which clustering is the best. There is however a cost function but it may be hidden. This chapter addresses several cost functions that are used in existing clustering methods.

4.1 TOTAL SQUARED ERROR (TSE)

Total squared error (TSE) is the objective function for most centroid-based clustering algorithms such as k-means, which is the sum of variances in individual clusters. Given data inputs x_i, i=1..N, centroids c_j, j=1..k, and labels of data l_i, i=1..N, l_i=1..k, TSE is defined as [6]:

$$TSE = \sum_{i=1}^{N} \left\| x_i - c_{l_i} \right\|^2 \tag{4.1}$$

Mean squared error (MSE) equals normalized TSE by the total number of objects. There is no difference between minimizing MSE and TSE.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \left\| x_i - c_{l_i} \right\|^2 \tag{4.2}$$

For a fixed number of clusters k, the best clustering is the one that provides minimum TSE. However, when the number of

clusters varies, the clustering that best fits the data cannot be concluded merely based on TSE because increasing k will always provide a smaller TSE. This would lead all points into their own clusters.

The TSE in equation (4.1) can be used only for the data that the centroid of a cluster can be calculated by averaging the objects in the cluster.

4.2 ALL PAIRWISE DISTANCES (APD)

This cost function considers all pairwise distances (APD) between the objects in a cluster. The centroid is not needed. Therefore, APD can be used for any type of data if the distance between every two objects is available. The criterion is defined as:

$$APD = \sum_{x_i, x_j \in C_l} \left\| x_i - x_j \right\|^2 \tag{4.3}$$

It can be shown for Euclidean distance that [33]:

$$\begin{aligned} APD = APD_1 + APD_2 + ... + APD_k = \\ n_1 TSE_1 + n_2 TSE_2 + ... + n_k TSE_k \end{aligned} \tag{4.4}$$

where APD_i, n_i, and TSE_i are the sum of all pairwise distances, the number of objects, and the total squared error in cluster i, respectively. It is shown in [34] that applying all pairwise distances as the clustering criterion leads to more balanced clusters than TSE.

TSE can be calculated for non-numeric data without having centroids as follows. The sum of all pairwise distances is calculated for each cluster i, and the result is divided by the number of objects in the cluster giving the total squared error TSE_i. Summing up the total squared errors of all clusters results in TSE.

4.3 SPANNING TREE (ST)

The cost function is the sum of the costs of *spanning trees* (ST) of the individual clusters. The optimal solution for the cost function is achieved from the minimum spanning tree (MST) of the data objects. Given the MST in Figure 4.1 (left), we can get three clusters by cutting the two largest links. This cost function is suitable for detecting well separated arbitrary shaped clusters. However, it fails in real life data sets with noise, see Figure 4.1 (right).

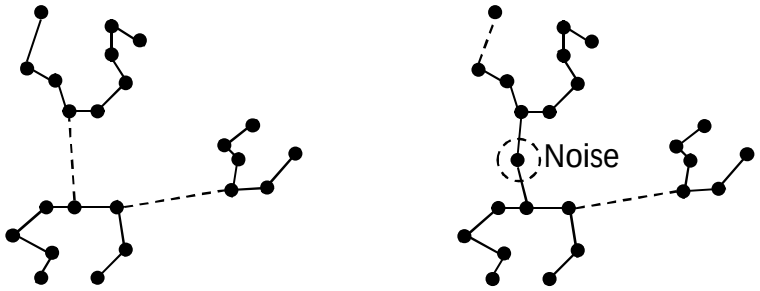


Figure 4.1: Spanning trees of clusters are used to derive the cost function.

4.4 K-NEAREST NEIGHBOR CONNECTIVITY

This cost function measures connectedness by counting the number of *k* nearest neighbors of each object that are placed in different cluster than the object [35]. It is calculated as:

$$K-CONN=\sum_{x_i\in P_l}\sum_{x_j\in nn(x_i)}\delta_{x_i}(x_j) \qquad \delta_{x_i}(x_j)=\begin{cases}\frac{1}{j}, & if\ x_j\notin P_l\\ 0, & otherwise\end{cases} \tag{4.5}$$

where x_j is the j^{th} nearest neighbor of x_i, and P_l represents the cluster that x_i belongs to. The number of neighbors *k* is an input parameter. The cost function should be minimized. The optimal case is when all *k* nearest neighbors of an object locate in the same cluster of the object. The impact of the first neighbor on the cost function is the highest, and it decreases for the next

neighbors by the factor 1/*j*, *j*=1..*k*. The 5 nearest neighbors of one object is depicted in Figure 4.2, from which the fourth and fifth neighbors are from the other cluster. The error is calculated as 1/4+1/5=0.45. Summing up the errors for all the points gives the value of cost function.

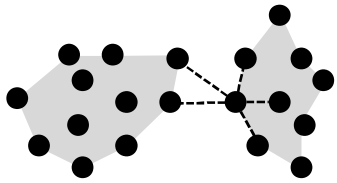


Figure 4.2: Five nearest neighbors are considered to calculate the cost function. For the selected point, two neighbors are located in the other cluster.

4.5 LINKAGE CRITERIA

In agglomerative clustering, a global cost function has not been defined in the literature. Instead, a merge cost is defined which aims at optimizing the clustering locally. Several criteria such as single-link and complete-link are used for merging two clusters, see Figure 4.3. We reveal the global cost function through analyzing the local ones.

Single-link criterion is the distance between the two most similar objects in two clusters. The goal of single-link is to find clusters with the highest connectivity. Two objects in a cluster can be far away but connected through other points in the cluster. The cost function is the sum of the costs of spanning trees of individual clusters. Single-link can be related to Kruskal's algorithm which is known to be optimal for MST. It can be shown that *k* clusters correspond to the MST forest of *k* trees.

Complete-link criterion is the distance between the two most dissimilar objects in two clusters. Complete-link aims at finding homogenous clusters so that the maximum distance between the objects in each cluster is minimized. Once two new clusters are merged, the resulting distance is the maximum distance over all

clusters which indicates the worst cluster. Given a clustering, the largest pairwise distance in each cluster is determined. The overall cost function is the maximum of the largest distances from all clusters. We call the cost function MAX-MAX. Agglomerative clustering using the complete-link criterion does not guarantee the optimal solution for the MAX-MAX cost, see Figure 4.4.

Average-link criterion selects the two clusters that the average distance between all pairs of objects in them is minimum. The corresponding cost function is therefore all pairwise distances.

Centroid-link criterion is the distance between the centroids of two clusters. It can be used only for data in which the centroids of clusters can be derived.

Ward's criterion selects the clusters to be merged that result in a minimum increase in TSE [36]. The increase of TSE resulted from merging two clusters *i* and *j* is calculated as:

$$\Delta TSE = \frac{n_i n_j}{n_i + n_j} \left\| c_i - c_j \right\|^2 \tag{4.6}$$

where c_i and c_j are the centroids, and n_i and n_j are the number of objects in the two clusters.

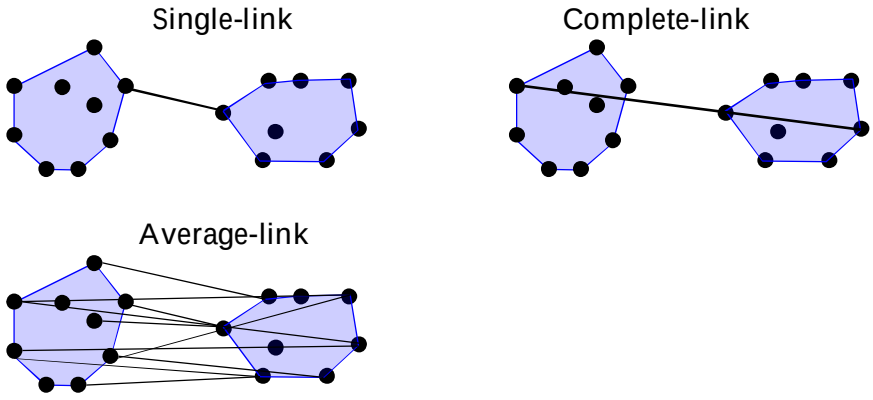


Figure 4.3: Distance between two clusters

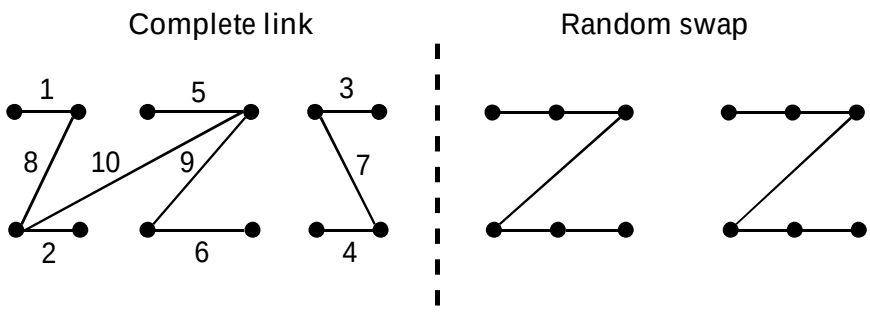


Figure 4.4: Complete link agglomerative clustering (left) results in a higher value of the cost function MAX-MAX comparing to the random swap algorithm (right). The numbers show the order of merges.

5 Internal validity indices

Clustering is defined as an optimization problem in which the quality is evaluated directly from the optimization criterion. Straightforward criterion works with a fixed number of clusters *k*. Internal validity indices extend this to variable *k*.

5.1 INTERNAL INDICES

Internal indices use a clustering and the underlying data set to assess the quality of the clustering [37]. They are designed based on the goal of clustering, placing similar objects in the same cluster and dissimilar objects in different clusters. Accordingly, two concepts are defined: intra-cluster similarity and inter-cluster similarity. Intra-cluster similarity (e.g. compactness, connectedness, and homogeneity) measures the similarity of the objects within a cluster, and inter-cluster similarity or separation measures how distant individual clusters (or their objects) are.

Compactness is suitable for the clustering algorithms that tend to provide spherical clusters. Examples include centroid-based clustering algorithms such as K-means, and average-link agglomerative clustering. Connectedness is suitable for density-based algorithms such as DBSCAN [37]. Several variants of compactness and connectedness exist. The average of pairwise intra-cluster distances and the average of centroid-based similarities are representatives of compactness. A popular measure of connectedness is k-nearest neighbor connectivity which counts violations of nearest neighbor relationships [37].

A good clustering of a data set is expected to provide well separated clusters [38]. Separation is defined in different ways. Three common methods are the distance between the closest objects, the most distant objects, and the centers of two clusters [39].

Several internal indices have been proposed that combine compactness and separation [3] [37] [39] [40] [41] [42]. Popular indices are listed in Table 5.1. Most of the indices have been invented for determining the number of clusters that fits the data.

Table 5.1: Selection of popular internal validity indices

SSW [43]	$\sum_{i=1}^{N} \left\| x_i - c_{l_i} \right\|^2$
SSB [43]	$\sum_{i=1}^{K} n_i \left\| c_i - \bar{x} \right\|^2$
Calinski-Harabasz [44]	$\frac{SSB/(K-1)}{SSW/(N-K)}$
Ball&Hall [45]	SSW / K
Xu-index [46]	$D \log_2(\sqrt{SSW/(DN^2)}) + \log K$
Dunn's index [47]	$\frac{\min_{i=1}^{M} \min_{j=i+1}^{M} d(c_i, c_j)}{\max_{k=1}^{M} diam(c_k)}$ where $d(c_i, c_j) = \min_{x \in c_i, x' \in c_j} \left\| x - x' \right\|^2$ and $diam(c_k) = \max_{x, x' \in c_k} \left\| x - x' \right\|^2$
Davies&Bouldin [48]	$\frac{1}{K} \sum_{i=1}^{K} \max_{j=1..M, j \neq i} R_{ij}$ where $R_{ij} = \frac{MSE_i + MSE_j}{\left\| c_i - c_j \right\|^2}$ and $MSE_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \left\| x_j - c_i \right\|^2$
SC [49]	$\frac{1}{N} \sum_{p=1}^{N} \frac{b(x_p) - a(x_p)}{\max(a(x_p), b(x_p))}$ where

	$a(x_p) = \frac{1}{n_i - 1} \sum_{q=1, j \neq i}^{n_i} \left\| x_p - x_q \right\|^2 \quad x_p, x_q \in C_i$ $b(x_p) = \min_{q=1}^{N} \left\| x_p - x_q \right\|^2$ $a(x_p) = \min_{q=1}^{N} \left\| x_p - x_q \right\|^2 \quad x_p \in C_i, x_q \notin C_i$
BIC [43]	$L * N - \frac{1}{2} K(D+1) \sum_{i=1}^{M} \log(n_i)$
Xie-Beni [50]	$\frac{\sum_{i=1}^{N} \sum_{j=1}^{K} u_{ij}^2 \left\| x_i - c_k \right\|^2}{N \min_{t \neq s} \{ \left\| c_t - c_s \right\|^2 \}}$
WB [51]	$\frac{K * SSW}{SSB}$

5.2 SUM OF SQUARES WITHIN CLUSTERS (SSW)

Sum of squares within clusters (SSW) [43] or within cluster variance is equal to the TSE, see Figure 5.1.

The index can only be used for numerical data because it requires centroids of clusters. SSW measures the compactness of clusters, and is suitable for centroid-based clustering, where hyperspherical clusters are desired. The value of SSW always decreases as the number of clusters increases.

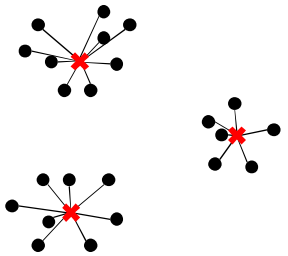


Figure 5.1: Illustration of the sum of squares within clusters

5.3 SUM OF SQUARES BETWEEN CLUSTERS (SSB)

The *sum of squares between clusters* (SSB) [43] measures the degree of separation between clusters by calculating between cluster variance.

The separation between clusters is determined according to the distances of centroids to the mean vector of all objects, see Figure 5.2. The factor n_i in the formula presented in Table 5.1 indicates that a cluster with a bigger size has more impact on the index. This criterion requires the centroids or prototypes of clusters and all data. Increasing the number of clusters usually results in a larger SSB value.

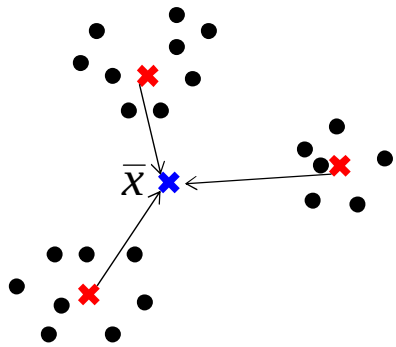


Figure 5.2: Illustration of the sum of squares between clusters.

5.4 CALINSKI-HARABASZ INDEX (CH)

The *Calinski-Harabasz* (CH) [44] index uses the ratio of separation and compactness to provide the best possible separation and compactness simultaneously. A maximum of the index value indicates the best clustering with a high separation and low error in compactness. A higher number of clusters for a data set provides higher SSB and lower SSW. However, the decrease in SSW is more than that of SSB. Therefore, the penalty factor (*K*-1) prevents the conclusion of a higher number of clusters than the correct one. The term *N-K* is considered to support cases in which the number of clusters is comparable to

the total number of objects. However, usually N is much higher than K, and the term can be shortened to N.

This index, similar to SSB and SSW, is limited to numerical data with hyperspherical clusters.

5.5 SILHOUETTE COEFFICIENT (SC)

Silhouette coefficient (SC) [49] measures how well each object is placed in its cluster, and separated from the objects in other clusters. The average dissimilarity of each object x_i with all objects in the same cluster is calculated as $a(x_i)$, which indicates how well x_i is assigned to its cluster. Lowest average dissimilarity of x_i to other clusters is calculated as $b(x_i)$.

$$\text{SC}=\frac{1}{N}\sum_{p=1}^{N}\frac{b(x_i)-a(x_i)}{\max(a(x_i),b(x_i))} \tag{5.1}$$

The dissimilarity between two objects is sufficient for calculating the index. Therefore, SC can be used for any type of data, and any clustering structure.

5.6 DUNN FAMILY OF INDICES

Dunn index [47] is defined as follows:

$$DI=\frac{\min\limits_{i=1}^{K}\min\limits_{j=i+1}^{K} d(c_i,c_j)}{\max\limits_{k=1}^{K} diam(c_k)} \tag{5.2}$$

where $d(c_i,c_j)$ is the dissimilarity between two clusters and $\text{diam}(c_k)=\max d(x_i, x_j)$ is the diameter of cluster c_k, where $x_i, x_j \in c_k$. The numerator of the equation is a measure of separation, the distance between the two closest clusters. The diameter of a cluster shows the dispersion (opposite to compactness) of the cluster. The cluster with the maximum diameter is considered. A larger value of the index indicates a better clustering of a data set with more compact and well separated clusters.

Dunn index is sensitive to noise, and has a high time complexity [52]. Three related indices have been introduced in [52] based on Dunn index to alleviate these limitations. They are called Dunn-like indices.

5.7 SOLVING NUMBER OF CLUSTERS

To determine the number of clusters, clustering is applied to the data set for a range of $k \in [K_{min}, K_{max}]$, and the validity index values are calculated. The best number of clusters k^* is selected according to the extremum of the validity index.

Figure 5.3 shows data set S_1 with 15 clusters and the normalized values of SSW and SSB. Random swap clustering algorithm [25] is applied when the number of clusters is varied in the range [2, 25].

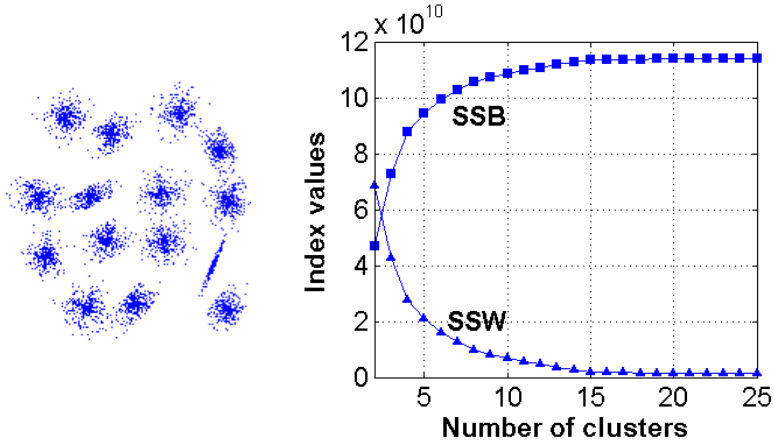


Figure 5.3: Data set S_1 (left), and the measured values of SSW and SSB (right)

The error in compactness measured by SSW decreases, and the separation measured by SSB increases, as the number of clusters increases. However, the decreasing and increasing rates significantly reduce after k=15, a knee point that indicates the correct number of clusters. Although several methods for detecting the knee point have been summarized in [43] but none of them work in all cases. It would be easier to use a validity

index that provides a clear minimum or maximum value at the correct number of clusters. For example, CH [44] provides a maximum by considering both SSW and SSB, and also a penalty factor on the number of clusters *k*, see Figure 5.4.

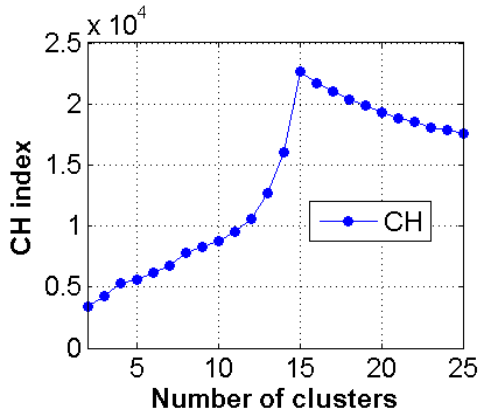


Figure 5.4: Determining the number of clusters for the data set S_1 using CH index

Most of the existing internal indices require the prototypes of the clusters but these are not always easy to calculate, such as in a clustering of words based on their semantic similarity. In [P4], we introduce a new internal index to be used for determining the number of clusters in a hierarchical clustering of words.

To find out which level of the hierarchy provides the best categorization of the data, an internal index needs to evaluate the compactness within clusters and separation between clusters at each level. We define the proposed index as the ratio of compactness and separation:

$$SC(k) = \frac{C(k)}{S(k)} \tag{5.3}$$

$$C(k) = \max_{t}\{\max_{i,j} JC(w_i, w_j), w_i \neq w_j \in c_t\} + I_1 / N \tag{5.4}$$

$$S(k) = \frac{\sum_{t=1}^{k}\sum_{s>t}^{k} \min_{i,j} JC(w_i, w_j), w_i \in c_t, w_j \in c_s}{k(k-1)/2} \tag{5.5}$$

where w_i is the i^{th} keyword, c_t is the cluster t at the level of hierarchy where the number of clusters is k, JC is the Jiang & Conrath function that measures the distance of two words, I_1 is the number of clusters with only one word, and N is the total number of words.

Compactness measures the maximum pairwise distance in each cluster, and takes the maximum value among all clusters. Compactness for clusters with a single object cannot be considered zero because the clustering in which each object is in its own cluster would then result in the best compactness. To avoid this, we add the factor I_1/N to the compactness equation. In the beginning of clustering, when each object belongs to its own cluster, the compactness equals 1 because $I_1=N$.

Separation measures the minimum distance between the words of every two clusters and sums up the values. Normalization by $k(k\text{-}1)$ provides a value in the same scale as compactness. A good clustering provides a small distance value for compactness and a large distance value for separation. Therefore, the level of the hierarchy with k clusters that results in the minimum SC is selected as the best level.

6 External validity indices

External validity indices measure how well the results of a clustering match the ground truth (if available) or another clustering [53] [P1]. They are the criteria for testing and evaluating clustering results and for the analysis of clustering tendency in a data set. Some authors define an external index for comparing a clustering with ground truth [4] [37] and define *relative index* for comparing two clusterings of a data set [3] [5]. However, many others classify both as external index. External indices have been used in ensemble clustering [40] [54] [55] [56], genetic algorithms [57], and evaluating the stability of k-means [55].

In this section, we first introduce several properties for a validity index based on which its performance can be evaluated. We then provide a review of the external indices in three categories: *pair-counting*, *information theoretic,* and *set-matching*, see Table 6.1, [P2]. Finally, we describe our new setup of experiments for evaluating the external indices.

Given two partitions $P=\{P_1, P_2,\ldots,P_K\}$ of K clusters and $G=\{G_1, G_2,\ldots,G_{K'}\}$ of K' clusters, an external validity index measures the similarity between P and G. Most external indices are derived using the values in the *contingency table* of P and G, see Table 6.2. The table is a matrix where n_{ij} is the number of objects that are both in clusters P_i and G_j: $n_{ij}=|P_i \cap G_j|$, n_i and m_j are the size of clusters P_i and G_j respectively.

Table 6.1: External validity indices

Pair-counting measures	
Rand index [58]	$RI = \frac{a+d}{N(N-1)/2}$
Adjusted Rand index [59]	$ARI = \frac{RI - E(RI)}{1 - E(RI)}$
Information theoretic measures	

Mutual information [60]	$MI = \sum_{i=1}^{K}\sum_{j=1}^{K'} \frac{n_{ij}}{N} \log \frac{N n_{ij}}{n_i m_j}$
Normalized mutual information [60]	$NMI_1 = \frac{MI(P,G)}{(H(P)+H(G))/2}$ $NMI_2 = \frac{MI(P,G)}{\sqrt{H(P)\times H(G)}}$
Normalized Variation of Information [61]	$NVI = \frac{H(P)+H(G)-2MI(P,G)}{H(P)+H(G)}$
Set-matching measures	
F measure [62]	$FM = \frac{1}{N}\sum_{i=1}^{K} n_i \max_j \frac{2n_{ij}}{n_i + m_j}$
Criterion H [63]	$H = 1 - \frac{1}{N}\max_j \sum_{i=1}^{K} n_{ij}$
Normalized Van Dongen [64]	$NVD = \frac{2N - \sum_{i=1}^{K}\max_{j=1}^{K'} n_{ij} - \sum_{j=1}^{K'}\max_{i=1}^{K} n_{ij}}{2N}$
Purity [5]	$Purity = \frac{1}{N}\sum_{i=1}^{K}\max_{\pi} n_{i,\pi}(i)$
Centroid index [P1]	$CI_1(P,G) = \sum_{i=1}^{K'} orphan(G_i)$ $CI_2(P,G) = \max(CI_1(P,G), CI_1(G,P))$
Centroid similarity index [P1]	$CSI = \frac{\sum_{i=1}^{K} n_{ij} + \sum_{j=1}^{K'} n_{ji}}{2N}$ *i*, *j*: indices of matched clusters
Centroid ratio [65]	$CR = 1 - \sum_{i=1}^{K} \gamma_i / K$ $\gamma_i = \begin{cases} 1 & unstable\ pair \\ 0 & stable\ pair \end{cases}$
Pair sets index [P2]	$\begin{cases} \frac{S - E(S)}{\max(K,K') - E(S)} & S \geq E(S), \max(K,K') > 1 \\ 0 & S < E(S) \\ 1 & K = K' = 1 \end{cases}$ $S = \sum_{i=1}^{\min(K,K')} \frac{n_{ij}}{\max(n_i, m_j)}$ *i*, *j*: indices of paired clusters

Table 6.2: Contingency table for two partitions P and G

	G_1	G_2	...	G_j	...	$G_{K'}$	$\sum$
P_1	n_{11}	n_{12}	...	n_{1j}	...	$n_{1K'}$	n_1
P_2	n_{21}	n_{22}	...	n_{2j}	...	$n_{2K'}$	n_2
...	...	...	...	...	...	...	...
P_i	n_{i1}	n_{i2}	...	n_{ij}	...	$n_{iK'}$	n_i
...	...	...	...	...	...	...	...
P_K	n_{K1}	n_{K2}	...	n_{Kj}	...	$n_{KK'}$	n_K
$\sum$	m_1	m_2	...	m_j	...	$m_{K'}$	N

6.1 DESIRED PROPERTIES

An external validity index needs to satisfy several properties to be consistent and comparable for different data sets and clustering structures.

Normalization transforms the index within a fixed range, for example [0, 1], which makes comparison easier for data sets of a different size and structure. Normalization is the most commonly agreed property in the clustering community [66], and is usually performed as:

$$I_d^n(P,G) = \frac{I_d - \min(I_d)}{\max(I_d) - \min(I_d)} \tag{6.1}$$

where $min(I_d)$ and $max(I_d)$ are the minimum and maximum values of I_d.

Index values are expected to be constant when different random clusterings are compared with a ground truth [59]. A random partition is created by selecting a random number of clusters of random size. The similarity between the random partition and the ground truth originates merely by chance. Take an example of Rand index: the value of the index for two random partitions is not a constant, and is in a narrower range of [0.5, 1] instead of [0, 1]. By *correction for chance* or *adjustment*, the expected value of an index $E(I)$ is transformed to zero (similarity) or one (dissimilarity) [59] [67]. Adjustment and normalization can be performed jointly as follows:

$$\text{Dissimilarity}: \quad I_d^{adj}(P,G) = \frac{I_d - \min(I_d)}{E(I_d) - \min(I_d)}$$
$$\text{Similarity}: \quad I_s^{adj}(P,G) = \frac{I_s - E(I_s)}{\max(I_s) - E(I_s)} \tag{6.2}$$

where the minimum (similarity) or maximum (dissimilarity) is replaced by the expected value $E(I)$.

Metric property has also been considered. Although a similarity/dissimilarity measure can be effective without being a metric [7], it is sometimes preferred. Considering dissimilarity index I and clusters P_1, P_2 and P_3, metric properties require [2] [68]:

1. Non-negativity: $I_d(P_1,P_2) \geq 0$
2. Reflexivity: $I_d(P_1,P_2)=0$ if and only if $P_1=P_2$
3. Symmetry: $I_d(P_1,P_2)=I_d(P_2,P_1)$
4. Triangular inequality: $I_d(P_1,P_2)+I_d(P_2,P_3) \geq I_d(P_1,P_3)$

A similarity metric satisfies the following [2]:

1. Limited Range: $I_s(P_1,P_2) \leq I_0<\infty$
2. Reflexivity: $I_s(P_1,P_2)= I_0$ if and only if $P_1=P_2$
3. Symmetry: $I_s(P_1,P_2)=I_s(P_2,P_1)$
4. Triangular inequality:

$$I_s(P_1,P_2)\times I_s(P_2,P_3) \leq I_s(P_1,P_3)\times(I_s(P_1,P_2)+I_s(P_2,P_3))$$

The triangular inequality for a similarity index I_s is derived here according to the corresponding inequality for a dissimilarity index which is defined as c/I_s (c>0). However, other forms of the inequality are possible by defining other dissimilarities such as $\max(I_s)-I_s$. It is trivial to show that if c/I_s (or $\max(I_s)-I_s$) is a dissimilarity metric, I_s is a similarity metric as well [2]. Hence, metric properties for a similarity index can be checked for its corresponding dissimilarity [P2].

Cluster size imbalance signifies that a data set can include clusters with large difference in their sizes. Some researchers argue that clusters with larger sizes have more importance than smaller clusters but we assume that each cluster has the same importance independent of its size. Invariance in the size of clusters is therefore another desired property of an index. The size of a data set should not affect the index either [P2].

An index should be independent of the number of clusters. Some indices such as *Rand index* (RI) [58] give higher similarity when more clusters [68]. An index should also be applicable for comparing two clusterings with different number of clusters.

Monotonicity is another required property. This property states that the similarity of two clusterings monotonically decreases as their difference increases [P2].

Once these desired properties are met, then index values for different data sets are on the same scale and comparable. For instance, if an index gives 90% and 70% similarities, 90% should represent higher similarity. However, this is true only if the index is independent of the data set and its clustering structure [P2].

6.2 PAIR-COUNTING INDICES

Pair-counting measures count the pairs of points on which two clusterings agree or disagree. For instance, if two objects in one cluster in the first partition are also placed in the same cluster in the second partition, then this is considered an agreement. Most existing external validity indices are classified in this group [P2]. Four values are defined: *a* represents the number of pairs that are in the same cluster both in *P* and *G*; *b* represents the number of pairs that are in the same cluster in P but in different clusters in *G*; *c* represents the number of pairs that are in different clusters in *P* but in the same cluster in *G*; *d* represents the number of pairs that are in different clusters both in *P* and *G*. Values *a* and *d* count agreements while values *b* and *c* count disagreements. Examples of each case are illustrated in Figure 6.1. The values of *a*, *b*, *c,* and *d* can be calculated from the contingency table [59] as follows:

$$a = \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}(n_{ij} - 1)$$

$$b = \frac{1}{2}(\sum_{j=1}^{K'} m_j^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2)$$

$$c = \frac{1}{2}\left(\sum_{i=1}^{K} n_i^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\right) \tag{6.3}$$

$$d = \frac{1}{2}\left(N^2 + \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2 - \left(\sum_{i=1}^{K} n_i^2 + \sum_{j=1}^{K'} m_j^2\right)\right)$$

Rand index [58], a well known pair-counting measure, equals the number of agreements divided by the total number of pairs of points:

$$I_d^n(P,G) = \frac{a+d}{a+b+c+d} = \frac{a+d}{N(N-1)/2} \tag{6.4}$$

For random partitions, the similarity between two clusterings is desired to be close to zero. However, the expected value of Rand index for random partitions is 0.5 and the index is within a narrow range of [0.5, 1] according to a number of studies [40] [55] [59]. Hence, a corrected-for-chance version called *adjusted Rand index* (ARI) was introduced in [59] which is upper bounded by one and lower bounded by zero. The expected value of the Rand index is estimated using the hyper-geometric distribution assumption in which the size and number of clusters are fixed [59].

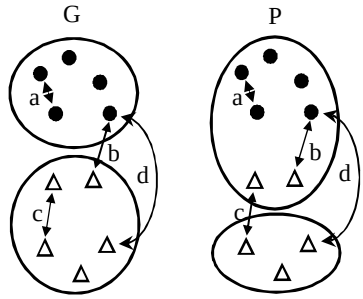


Figure 6.1: The principle of pair-counting measures.

6.3 INFORMATION-THEORETIC INDICES

Existing information theoretic measures employ the concept of entropy [60] to compare two partitions. A systematic study of this group, including several existing popular measures and

recently proposed measures, has been performed in [66]. Entropy is measured by the average number of bits needed to store or communicate data. The entropy of clustering P with K clusters is defined as:

$$H(P) = -\sum_{i=1}^{K} p(P_i) \log p(P_i) \tag{6.5}$$

where $p(P_i)=n_{i.}/N$ is the estimated probability of the cluster P_i.

With clustering G and the joint distribution $p(P,G)$, the average number of bits for P is derived by conditional entropy [53] as follows:

$$H(P|G) = \sum_{i=1}^{K} \sum_{j=1}^{K'} p(P_i, G_j) \log p(P_i | G_j) \tag{6.6}$$

where the probability $p(P_i,G_j)$ can be estimated from the contingency table as n_{ij}/N.

Mutual information (MI) [54] [66] is derived from conditional entropy and represents the similarity between two clusterings [68]. If we choose a random object in the data set, knowing its cluster in G, mutual information measures the reduction in uncertainty of the object's cluster in P [68] [69]. Mutual information is defined formally as follows:

$$MI(P,G) = H(P) - H(P|G) = H(P) + H(G) - H(P,G) \tag{6.7}$$

In terms of probabilities, it is:

$$MI(P,G) = \sum_{i=1}^{K} \sum_{j=1}^{K'} p(P_i, G_j) \log \frac{p(P_i, G_j)}{p(P_i)\, p(G_j)} \tag{6.8}$$

Variation of Information (VI) [69] is complementary of the mutual information, see Figure 6.2, and is calculated by summing up the conditional entropies $H(P|G)$ and $H(G|P)$:

$$\begin{aligned} VI(P,G) &= H(P|G) + H(G|P) = \\ &H(P) + H(G) - 2MI(P,G) = \\ &2H(P,G) - H(P) - H(G) \end{aligned} \tag{6.9}$$

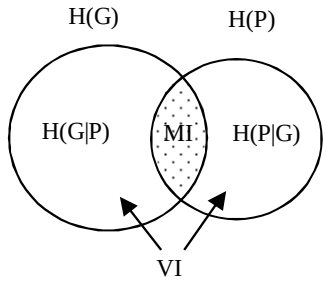


Figure 6.2: Mutual information and variation of information

Both MI and VI are metric but are not bounded to a fixed range [68]. The mutual information of clusterings *P* and *G* is lower bounded by zero. The geometric or arithmetic mean of entropies as an upper bound can be an option for normalization [54] [60] [68], see Table 6.1. In [60], *min*(*H*(*P*), *H*(*G*)) and *max*(*H*(*P*), *H*(*G*)) are also used for normalization. An upper bound for VI is *H*(*P*)+*H*(*G*), which means that clusterings *P* and *G* do not share any information [61]. The upper bound can therefore be used for the normalization of VI. In [P2], we prove that under the hyper-geometric distribution assumption and by using *H*(*P*)+*H*(*G*) for normalization, the adjusted forms of MI and VI are equal to their normalized forms:

$$AVI_s = NVI_s = AMI = NMI \tag{6.10}$$

where NVI_s and AVI_s denote the similarity form of NVI and AVI (1-NVI and 1-AVI) respectively.

6.4 SET MATCHING INDICES

Set-matching based indices are based on pairing similar clusters in two partitions. Taking use of the tight connection between partitions and centroids, cluster-level similarity indices employ representatives of clusters instead of point-level partitions.

Point-level indices consider the intersection of paired clusters in two clusterings. Examples of point-level set-matching measures are: *Purity* [5], *F-measure* (FM) [62], *Criterion H* (CH) [63], *normalized*

Van Dongen (NVD) [64], *centroid similarity measure* (CSI) [P1], and *Pair sets index* (PSI) [P2].

Cluster-level indices include *Centroid Index* (CI) [P1] and *Centroid Ratio* (CR) [65]. They only use cluster prototypes in contrast to point-level indices which employ the labels of all objects in resulting partitions.

Set-matching measures involve three design questions:

1. How to measure the similarity of two clusters?
2. How to match the clusters?
3. How to calculate overall similarity?

Normalization and correction for chance (if applied) are also essential parts of overall similarity derivation. We next study all these questions including the normalization.

1. Similarity of two clusters

Let P_i and G_j be two clusters in P and G respectively. Most set-matching measures use $|P_i \cap G_j|$ to calculate the similarity of the two sets. For example, in Figure 6.4, clusters G_1 and P_1 are more similar than G_2 and P_2 since the number of shared objects is 6 and 4 respectively. Many other ways to measure the similarity of two sets exist in the literature [70] and any of them can be employed for calculating the similarity of two clusters. Three popular measures are *Jaccard* (J) [71], *Sorensen-Dice* (SD) [72], and *Braun-Banquet* (BB) [70].

$$J = \frac{|P_i \cap G_j|}{|P_i \cup G_j|} \tag{6.11}$$

$$SD = \frac{2|P_i \cap G_j|}{|P_i| + |G_j|} \tag{6.12}$$

$$BB = \frac{|P_i \cap G_j|}{\max(|P_i|, |G_j|)} \tag{6.13}$$

Distance forms of J and SD are defined as (1-J) and (1-SD) where the former is a true metric but the latter does not satisfy triangular inequality. To make the measure independent of

cluster size, these measures normalize the number of shared objects $|P_i \cap G_j|$ in three different ways [P2].

FM [68] uses *precision* and *recall* concepts by measuring n_{ij}/n_i and n_{ij}/n_j respectively. The criterion [2×precision×recall/ (precision+recall)] would be equivalent to SD but avoids normalization by cluster size using n_i×SD instead of SD. PSI uses BB, and other point-level indices use the number of shared objects [P2]. Cluster-level indices provide a binary result (0 or 1) indicating whether the clusters have a 1:1 match (CI), or the pair of clusters is unstable (CR).

2. Matching

For every cluster, the pair to which the similarity is measured needs to be found. Three cases are considered: optimal pairing, greedy pairing, and matching. Matching is performed based on nearest neighbor mapping so that any cluster in *P* is matched to a cluster in *G* with maximal similarity. Several clusters can be matched with the same cluster in the other clustering. Pairing is a special case of matching in which clusters are only allowed to be matched once.

Matching results, in general, are not symmetric when finding pairs for clusters of *P* from *G* and vice versa. To make the index symmetric, similarity results in both directions are usually combined, see NVD, CI, and CSI equations in Table 6.1. FM and Purity assume the comparison of a clustering with ground truth and therefore consider matching in one direction only. The matching criterion in NVD and Purity is the number of shared objects; CI and CSI are based on the similarity of prototypes.

The pairing problem, however, is not trivial to solve and different algorithms have been proposed to find approximate or optimal solutions. Pairing can be seen as a matching problem in a weighted bipartite graph where nodes represent the clusters, see Figure 6.3. Greedy pairing is mostly used with the time complexity of $O(N^2)$. The two most similar clusters are iteratively matched and excluded. CH and CR use greedy pairing whereas PSI uses optimal pairing by Hungarian algorithm with time complexity $O(N^3)$, where N is the maximum number of clusters in *P* and *G*.

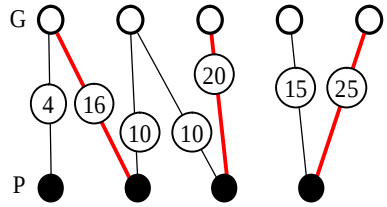


Figure 6.3: Pairing clusters to maximize overall similarity. The thick lines show the optimal pairing where the overall similarity according to number of shared objects would be (25+20+16)=61.

Figure 6.4 demonstrates the matching from *G* to *P* based on the number of shared objects where P_2 remains unmatched. Matching from *P* to *G* will be different resulting in (P_1,G_1), (P_2,G_2), and (P_3, G_3).

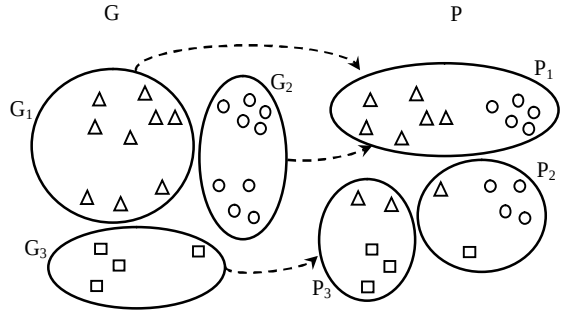


Figure 6.4: Matching clusters based on maximum shared objects. Cluster P_2 remains unmatched. In the pairing process of CH, G_2 is paired with P_2 after excluding G_1 and P_1 as the first pair.

Figure 6.5 shows matching in CI when there is different number of clusters. In matching *P* to *G*, one orphan centroid is found that indicates one difference in the global allocation of the clusters. In comparing two clusterings with different numbers of clusters, unpaired clusters indicate a disagreement in the number of clusters, which is an advantage of pairing.

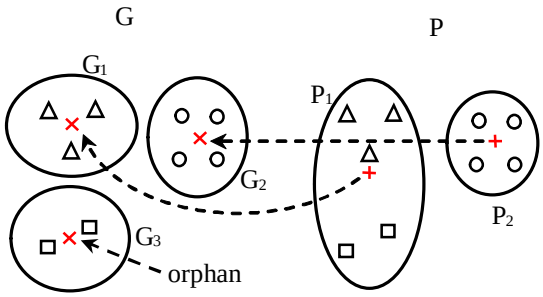


Figure 6.5: Matching centroids from *P* to *G* based on nearest neighbor mapping used in CI and CSI; one orphan centroid shows one difference in global allocation.

3. Overall similarity

Overall similarity is obtained by summing up the similarities of all the matched clusters. The upper bound of overall similarity for CH is *N* (total number of objects) which is used for normalization, see Table 6.1. To remove the asymmetric effect of matching, NVD and CSI use 2*N* because of two-way matching, see Table 6.1. In [P2], we show that CSI, Purity, NVD, and CH are all equivalent if their matching results are the same.

The overall dissimilarity of CI equals the number of zero mapped centroids of *G*. In Figure 6.6, the blue prototypes are mapped to the red prototypes from another solution according to minimum Euclidean distance. There is no mapping to two of the red prototypes, which results in CI=2. Since CI is not symmetric, CI_2 is defined as max(CI(*P*,*G*), CI(*G*,*P*)) [P1]. Centroid index represents the number of differences in global allocations and is in the range of [0, *K*-1], where *K* is the maximum number of clusters in the two clusterings. At least one non-zero mapped centroid exists, therefore the upper bound becomes *K*-1.

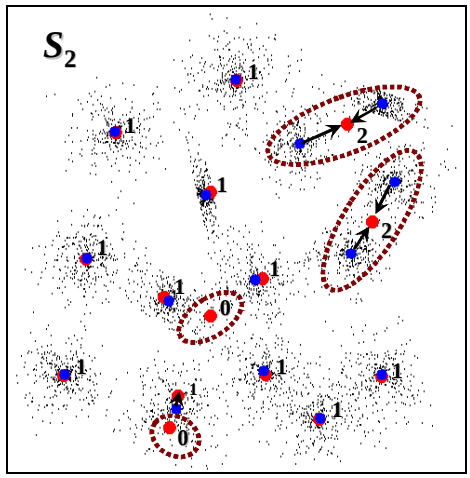


Figure 6.6: Two sets of prototypes and their mappings are shown. There are two orphans resulting in the index value of CI=2.

Centroid ratio (CR) defines the concept of (un)stable centroids. Consider a paired centroid C_i and C'_j with distance D_{ij} from clusterings P and G, respectively. Assume that the distances of C_i to the nearest centroid in P, and C'_j to the nearest centroid in G, are D_i and D_j. Then, if $D_{ij}^2/(D_i \times D_j) > 1$, the pair is considered unstable. The overall similarity is defined based on the number of unstable pairs [65], see Table 6.1.

In [P2], we propose pair sets index that is the only set-matching based index that applies correction for chance. We show that the simplified variant of PSI holds all the requirements to be a metric.

6.5 EXPERIMENTAL SETUP FOR EVALUATION

Partitions from real data sets provide only limited variations, whereas a variety of partitions with different data sizes, cluster sizes, and number of clusters should be used to provide a valid evaluation of the performance of an external index. In [P2], we introduce a new arrangement for experiments based on artificially generated partitions to investigate the properties of

external indices. First, we introduce the process of generating partitions, and then, we provide two examples that show the behavior of several external indices in two aspects: random partitions and monotonicity.

Consider a ground-truth partition *G* with 3,000 objects and 1,000 objects in each cluster, see Figure 6.7, where light grey, grey, and black represent the three clusters. In practice, we make an array of the length 3,000 objects with values 1, 2, and 3 representing cluster labels of data. In this case, the first 1,000 objects (light grey) have value 1. The partition *P* to be compared with is varied in different ways. The order of the data objects in the two partitions remains the same.

Figure 6.7: Two partitions with 3,000 objects.

Two partitions can be built in different ways to examine the properties of an external index with respect to different aspects.

1. Random partitions

Consider a partition *P* which consists of random labels as shown in Figure 6.8. Experiments are conducted for different numbers of clusters from *K*=1 to 20 in *P*. The indices NMI, ARI, and PSI give values close to zero independent of the number of clusters. The values of the other three indices are not zero because they are not corrected for chance, see Figure 6.9. Normalized mutual information gives zero in this case which shows that NMI has the same performance as the adjusted mutual information. This result further verifies the claim made in (6.10).

Figure 6.8: Clustering *P* is a random partition with two clusters.

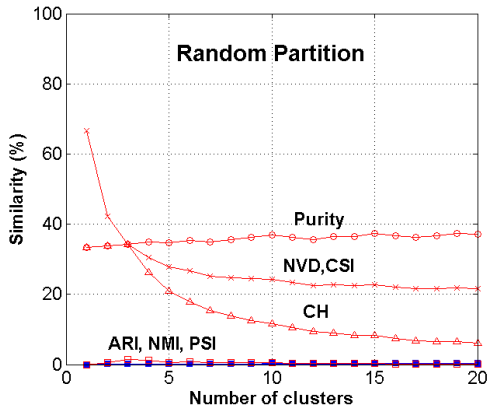


Figure 6.9: Random partitioning with different numbers of clusters in *P* from *K*=1 to 20

2. Monotonicity

The first (light grey) cluster in *P* is enlarged in steps of 50 objects until only one cluster remains, see Figure 6.10. In Figure 6.11, NMI, ARI, and NVD have very clear knee points when the light grey cluster reaches 2,000 objects because, at this point, the number of clusters decreases by 1. For NMI and ARI, the index values increase when the cluster size approaches 2,000. In this situation, there are still three clusters and the results indicate that NMI and ARI ignore relatively small clusters and weigh large clusters more. When the size of the light grey cluster is passing from 2,000, there is a local maximum as the number of clusters changes from three to two. NVD is constant between 1,500 to 2,000, and 2,500 to 3,000. The asymmetric matching of clusters in *NVD* causes the problem. Suppose that the size of the grey cluster (*x*) in *P* is less than 500. The number of shared objects is 1,000+*x*+1,000 in matching *P* to *G*. In matching *G* to *P*, both light grey and grey clusters in *G* are matched with the light grey cluster in *P*, resulting the number of shared objects 1,000+(1,000-*x*)+1,000. Summing up, the number of shared objects in two directions is independent of *x* and equal to 5,000. Therefore, when the size of the first cluster is between 1,500 and 2,000, the similarity remains a constant 5,000/6,000=0.83.

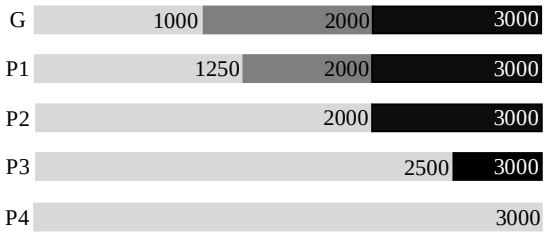


Figure 6.10: Enlarging the first (light grey) cluster in steps of 50 objects by moving the objects from the other two clusters

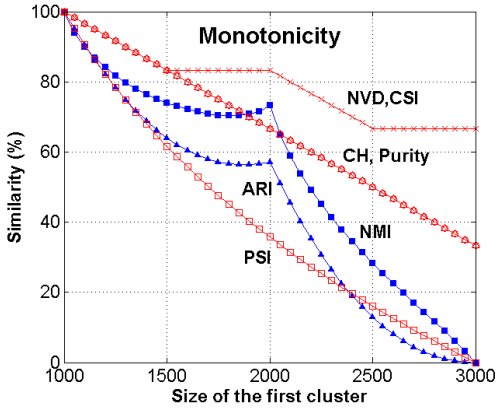


Figure 6.11: Increasing the size of the first cluster until it contains all data objects

6.6 SOLVING THE NUMBER OF CLUSTERS

External indices have been used for determining the number of clusters [4] [41] [73] [74] [75] [76] [77]. The idea is to generate randomness in the process by resampling the data, cluster the subsamples with a varying number of clusters, and then measure the stability with the presence of the randomness [74]. Stability is measured by comparing clusterings in the resamples using an external index. All existing methods under different nomenclature such as cross-validation [78], replication [77] [79], resampling [4] [74] [80] and prediction [73] [81], evaluate the stability of clustering results.

The idea is demonstrated in Figure 6.12. Centroid-based clustering is applied to the data set with five clusters and its subset for k=5 and k=8. The clustering results of the data set and the subset are similar when k=5, whereas there are disagreements when k=8. There are pairs of objects that are in the same cluster in the data set but in different clusters in the subset.

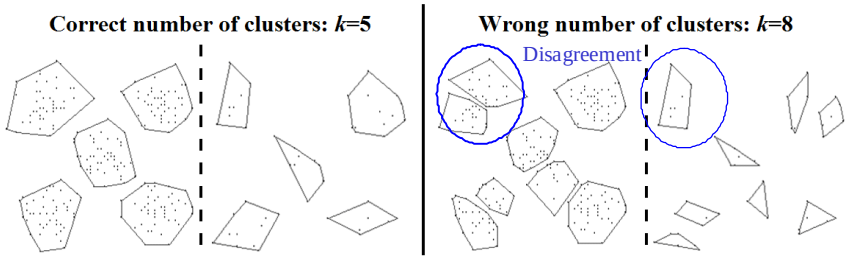


Figure 6.12: Stability-based method for finding the number of clusters. Stable (left) and unstable (right) results are produced when the correct and incorrect number of clusters are applied.

Stability, however, can be achieved with fewer clusters if the positioning of the clusters is not symmetric [82]. Figure 6.13 demonstrates two data sets with three well-separated clusters, first with a symmetric (left), and second with a non-symmetric (right) positioning of clusters. Applying clustering for k=2 gives stable results for the first data set and unstable results for the second data set. The second data set is also stable for k=3, which is the correct number of clusters. Therefore, it is better to select the highest number of clusters that leads to a stable result.

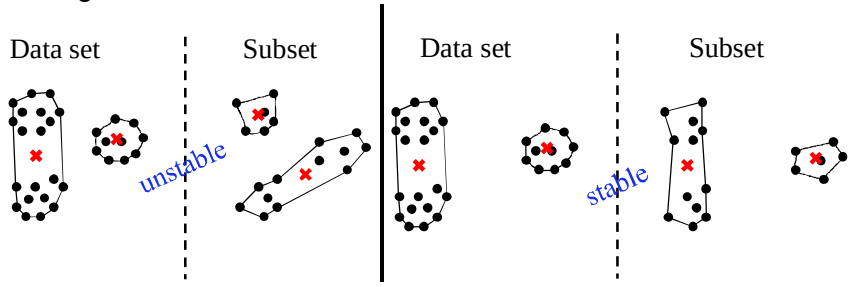


Figure 6.13: Unstable results for symmetrically and stable results for non-symmetrically positioned clusters when the incorrect number of clusters k=2 is applied.

The stability-based method includes four main design choices:

1. Adding randomness
2. Cross-validation strategy
3. Selection of the external index
4. Selection of the clustering method

Randomness is typically created by sub-sampling. The size and number of subsamples are parameters. Another approach is to use a randomized algorithm [83]. However, an inconsistent clustering algorithm such as k-means is completely unreliable and should not be used, but randomizing another more stable algorithm could be used. Adding noise has also been used to provide randomness in the data [84], [85]. A noise vector with random orientation can be generated but its magnitude depends on data and is not trivial to set. In the case of categorical data, adding noise can become complicated. Changing just one attribute randomly may result in an impossible combination of the attributes.

Most external indices are restricted to compare partitions of the same data exactly. A straightforward approach [41] [42] [74] compares clustering results to the result of the full set, but restricting only to the points that are in the subset. Another approach predicts the missing partition labels by nearest neighbor mapping using cluster centroids, or by applying a more complicated classifier process [73] [78] [80] [86]. We will also consider comparing the subsets directly by using centroid index [P1], which does not require the partition of the data.

The third design choice is the selection of an external validity index. We show by experiments in [P3] that the exact choice of the measure is not important, but how it is applied matters. All existing stability-based methods select the number of clusters that provide maximum stability, but simple counter-examples show how it will fail. We therefore introduce an alternative hypothesis that several numbers of clusters can provide stable results, and choosing the maximum number of clusters among these is more reliable.

The last design choice is the selection of a clustering algorithm. K-means is commonly chosen but it is highly unstable itself and not useful. Another more robust algorithm, such as agglomerative clustering [87], random swap [25] or genetic algorithm [57], should be used instead. However, the main question is not which algorithm but rather which cluster model (cost function). If we apply squared error criterion but the data is not spherical, a clustering may be resulted that does not fit the data. Nevertheless, we should still be able to find the number of clusters that best fits to this model.

The baseline variant of cross validation using the sub-sampling strategy is outlined in Figure 6.14.

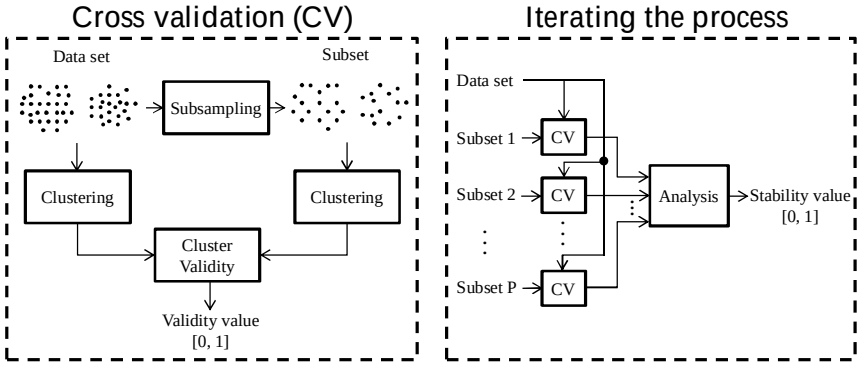


Figure 6.14: Cross-validation technique; clustering of a full data set is compared with the clustering of its subset (left). The process is repeated for a number of subsets (right).

The cross-validation approach is repeated by applying clustering with all potential numbers of clusters $k \in [k_{min}, k_{max}]$. We denote the mean value of the validity index for k clusters as I_k. Maximum stability approach uses this mean value as such to indicate the correct number of clusters:

$$K = \arg\max_{k}(I_k) \tag{6.14}$$

The *normalized maximum stability* approach selects the number of clusters as the maximum difference in mean stability values of the data (I) and the corresponding value (I_0) of the null

reference, which is a random data set drawn from the original data [41] [73]:

$$K = \arg\max_{k}(I_k - I_k^0) \tag{6.15}$$

This approach is referred to as normalization with regard to the number of clusters [83]. The reason is that the stability value depends on *k* regardless of the underlying data structure. For example, the stability of clustering for a random uniform data set decreases as the number of clusters increases. This bias should be removed, and then the same equation (6.14) should be used.

In [P3], we consider *last local maximum* as a new criterion, which provides better results. For this, a threshold (I_{th}) is set to decide how high of an index value is considered stable. The selection becomes:

$$K = \arg\max_{k}(k | I_k > I_{th}) \tag{6.16}$$

Resampling techniques have been used in supervised learning to improve prediction accuracy, where the main idea is that small changes in the training data will yield the same stable classifier without any significant change in accuracy. The same idea has been applied for estimating the number of clusters in a data set [80]. Part of the data is considered for training a classifier and the rest of the data for test. Two different labeling are derived for the test data: one from the classifier and the other by applying clustering. The two resulting partitions are compared using an external index, see Figure 6.15.

Figure 6.16 shows the results of cross-validation and classification-based approaches with and without normalization for the data set in Figure 6.12. The highest stability is found with *k*=5, the correct number of clusters.

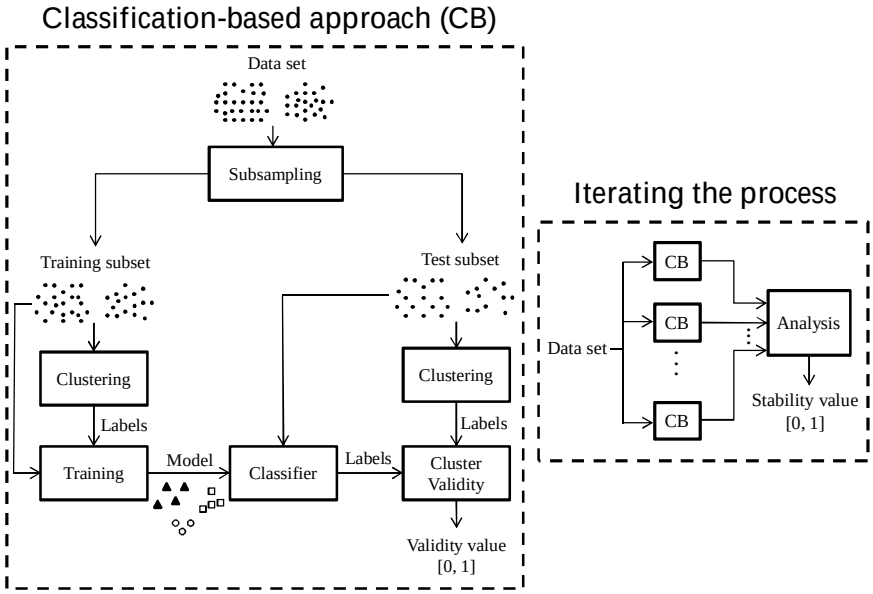


Figure 6.15: Classification-based approach (left), and iterating the process for several train and test sets (right).

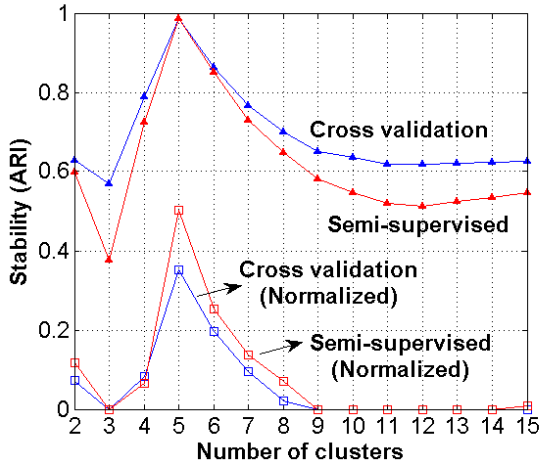


Figure 6.16. Example of stability-based method for the data set in Figure 6.12. 100 subsets are used in the cross-validation approach, each 20% of the full data set. The sizes of train and test sets in the classification-based approach are 80% and 20%. Random swap algorithm is used for clustering [25] and adjusted Rand index for validation [59].

7 Summary of contributions

This chapter summarizes the contributions of the five publications. Publications [P1] to [P4] concerns cluster validity, and publication [P5] proposes a semantic similarity measure for comparing groups of words.

In [P1], we propose a new cluster-level external validity index, which measures the global allocation of clusters instead of point-level differences in partitions. The proposed centroid index (CI) uses the representatives of the clusters to compare two clusterings, therefore it can be computed fast in $O(K^2)$ time. It is simple to implement, and has clear intuitive interpretations. Values CI>0 indicate how many clusters are differently allocated. Point-level extension of CI is also introduced. It belongs to the class of set matching-based indices. Experiments show that CI is capable of recognizing structural similarity of clusterings, even for high dimensional data. The results are also promising for solving the number of clusters based on measuring the stability of clusterings.

In [P2], we provide a systematic study of existing set matching-based external validity indices by analyzing three design questions: matching clusters, similarity of two clusters, and overall similarity. We show that how CSI, NVD, CH and purity are equivalent if the matching of clusters is the same. We study correction for chance, and prove that normalized mutual information and variation of information are intrinsically corrected for chance. We propose a new set matching based index called Pair Sets Index (PSI), which outperforms popular existing external indices. A novel setup for experiments is introduced based on synthetic data, which allows systematic evaluation of an external index for clusterings of different data sizes, cluster sizes, and numbers of clusters.

In [P3], we analyze the stability-based approach for determining the number of clusters. The goal is to find out

whether stability-based method can be used for determining the number of clusters. The simple answer is that, yes, it is possible, but we think it is not practical. If it is going to be used, we give the following recommendations how to construct the method. The exact choice of the cross-validation strategy and external index is not critical. Unstable clustering algorithms like k-means should not be used. Using the last local maximum criterion provides much better results than the global maximum criterion. Even if we demonstrated the approach working successfully for several data sets, we do not recommend it. External indices simply do not offer anything more that the best internal indices cannot offer, and they would just add unnecessary complications into the system.

In [P4], we propose a validity index for determining the number of clusters in a group of English words. We define compactness and separation between clusters, and the validity index as the ratio of compactness/separation. The experiments on a real data set show that the number of clusters calculated using the proposed index has a 2% error comparing to human judgment. The index uses only the similarity between two data objects, and therefore, is suitable for any type data.

In [P5], we propose a semantic similarity measure for comparing two groups of words. The measure is used for keyword-based clustering, where the objects such as documents, websites, and movies are represented by their keywords. We use Wu & Palmer index, a WordNet based measure, for comparing every two words. The proposed index is based on matching the words in two groups. A comparative evaluation with a real data set shows that the index avoids the limitations of traditional measures such as minimum or average similarity. The index can be used not only for comparing groups of words but for groups of any type of data, when the similarity between every two data objects is available.

8 Conclusions

The absence of prior information in cluster analysis makes it more challenging than supervised classification. The goal of cluster analysis is to reveal the underlying structure of the data rather than establishing classification rules. Cluster analysis contains a set of components including proximity measure, cost function, clustering algorithm, and cluster validity. Every component is closely related to the other components. Therefore, to analyze one component, knowledge of the other components and their effects is necessary. Given the same data set, different proximity measures, cost functions, and clustering algorithms usually result in different partitions.

This thesis reviews different components in cluster analysis, concentrated on cluster validity. Several novelties are presented such as proposing an internal index for determining the number of clusters in clustering of a group of words, introducing a cluster-level external validity index, proposing a point-level external validity index, providing an analysis of external indices and their properties, a novel setup of experiments for evaluating external indices, proposing a similarity measure for the comparison of two groups of words, and analysis of stability-based method for determining the number of clusters.

Though we have already seen many examples of successful applications of cluster analysis, many open problems still remain due to the existence of many inherent, uncertain factors. Our future research will entail:

- CI is limited to data for which centroid can be calculated. We can remove this dependency as long as the cluster similarity can be measured. This can be done point-wise but the overall idea of measuring the differences by the number of mismatch clusters is worth to try.
- Keyword clustering can be applied to clustering documents, for instance, web pages.

- Although we do not recommend the stability-based method for solving the number of clusters, we can use it for measuring stability of different algorithms and cost functions.
- Studying the cost functions and their properties should also be done. Analyzing what the different link and cut-based clustering methods actually optimize would reveal further insight.

References

[1] P. Perkhin, "A survey of clustering data mining techniques," *Grouping multidimensional data. Springer Berlin Heidelberg*, pp. 25-71, 2006.

[2] S. Theodoridis and K. Koutroumbas, "*Pattern Recognition*," 4th edn, Academic Press, New York, 2009.

[3] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "On clustering validation techniques," *J. Intelligent Information Systems*, 17(2-3), pp. 107–145, 2001.

[4] V. Roth, T. Lange, M. Braun, and J. Buhmann, "A resampling approach to cluster validation," *Compstat Physica-Verlag HD*, pp. 123-128, , 2002.

[5] E. Rendon, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. J. Computers and Communications*, 5(1), pp. 27-34, 2011.

[6] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks, 16*(3), pp. 645-678, 2005.

[7] A.A. Goshtasby, "Similarity and dissimilarity measures," in *Image Registration - Principles, Tools and Methods. Advances in Computer Vision and Pattern Recognition*, pp. 7-66, Springer London, 2012.

[8] R.W. Hamming, "Error detecting and error correcting codes," *Bell System technical journal*, *29*(2), pp. 147-160, 1950.

[9] J.C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, pp. 857-871, 1971.

[10] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, *33*(1), pp. 31-88, 2001.

[11] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," *AAAI*, 6, 2006.

[12] Y. Matsuo, T. Sakaki, K. Uchiyama, and M. Ishizuka, "Graph-based word clustering using a web search engine," *Conf. Empirical Methods in Natural Language Processing, Association for Computational Linguistics*, 2006.

[13] R.L. Cilibrasi and P. Vitanyi, "The google similarity distance," *IEEE Trans. Knowledge and Data Engineering*, 19(3), pp. 370-383, 2007.

[14] D. Bollegala, Y. Matsuo, and M. Ishizuka, "A web search engine-based approach to measure semantic similarity between words," *IEEE Trans. Knowledge and Data Engineering*, 23(7), pp. 977-990, 2011.

[15] L. Wu, X.S. Hua, N. Yu, W.Y. Ma, and S. Li, "Flickr distance: a relationship measure for visual concepts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(5), pp. 863-875, 2012.

[16] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, 32(1), pp. 13-47, 2006.

[17] I. Kaur and A.J. Hornof, "A comparison of LSA, WordNet and PMI-IR for predicting user click behavior," *SIGCHI*, *ACM Conf. Human factors in computing systems,* 2005.

[18] A. Gledson and J. Keane, "Using web-search results to measure word-group similarity," *Int. Conf. Computational Linguistics, Association for Computational Linguistics*, 1, 2008.

[19] B. Markines, C. Cattuto, F. Menczer, D. Benz, A. Hotho, and G. Stumme, "Evaluating similarity measures for emergent semantics of social tagging," *ACM Conf. World wide web*, 2009.

[20] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Short text clustering by finding core terms," *Knowledge and information systems*, 27(3), pp. 345-365, 2011.

[21] D. MacKay, "An example inference task: clustering," *Information Theory, Inference and Learning Algorithms*, *Cambridge: Cambridge university press*, pp. 284-292, 2003.

[22] N. Shi, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," *IEEE Int. Symp. Intelligent Information Technology and Security Informatics (IITSI)*, pp. 63-67, 2010.

[23] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," *ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics*, pp. 1027-1035, 2007.

[24] P. Fränti, "Genetic algorithm with deterministic crossover for vector quantization," *Pattern Recognition Letters*, 21(1), pp. 61-68, 2000.

[25] P. Fränti and J. Kivijärvi, "Randomised local search algorithm for the clustering problem," *Pattern Analysis and Applications*, 3(4), pp. 358-369, 2000.

[26] P. Fränti, O. Virmajoki, and V. Hautamäki, "Fast agglomerative clustering using a *k*-nearest neighbor graph," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11), pp. 1875-1881, 2006.

[27] C. D. Manning, P. Raghavan, and H. Schütze, "*Introduction to information retrieval,*" *Cambridge: Cambridge university press*, pp. 377-400, 2008.

[28] P. Fränti, T. Kaukoranta, D.-F. Shen, and K.-S. Chang, "Fast and memory efficient implementation of the exact PNN," *IEEE Trans. Image Processing*, 9(5), pp. 773-777, 2000.

[29] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *Int. Conf. Knowledge discovery and data mining*, pp. 226-231, 1996.

[30] M. Ankerst, M. Breunig, H. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Int. Conf. Management of data,* pp. 49-60, 1999.

[31] B. Liu, "A Fast Density-Based Clustering Algorithm for Large Databases," *Int. Conf. Machine Learning and Cybernetics*, pp. 996-1000, 2006.

[32] L. Zhao, J. Yang, and J. Fan, "A fast method of coarse density clustering for large data sets," *IEEE Int. Conf. Biomedical Engineering and Informatics (BMEI'09)*, pp. 1-5, 2009.

[33] H. Späth, "*Cluster analysis algorithms for data reduction and classification of objects,"* Wiley, New York, 1980.

[34] M. Malinen, "*New alternatives for k-means clustering*," PhD. thesis, Dept. Computer Science, University of Eastern Finland, 2015.

[35] J. Handl and J. Knowles, "Exploiting the trade-off—the benefits of multiple objectives in data clustering," In *Evolutionary Multi-Criterion Optimization,* Springer Berlin Heidelberg, pp. 547-560, 2005.

[36] H. Ward, "Hierarchical grouping to optimize an objective function," *J. American statistical association*, *58*(301), pp. 236-244, 1963.

[37] J. Handl, J. Knowles and D.B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, 21(15), pp. 3201-3212, 2005.

[38] M. Halkidi, Y. Batistakis, and M. Vazirgiannis "Cluster validity checking methods: part II," *SIGMOD Rec.*, 31(3), pp. 19-27, 2002.

[39] F. Kovács, C. Legány, and A. Babos "Cluster validity measurement techniques," In *Int. symp. hungarian researchers on computational intelligence*, 2005.

[40] S. Zhang, H. Wong and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognition*, 45(6), pp. 2214-2226, 2012.

[41] Q. Zhao, M. Xu, and P. Fränti, "Extending external validity measures for determining the number of clusters," *Int. Conf. Intelligent Systems Design and Applications (ISDA),* pp. 931-936, 2011.

[42] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics*, 19(4), pp. 459-466, 2003.

[43] Q. Zhao, "*Cluster validity in clustering methods*," PhD. thesis, Dept. Computer Science, University of Eastern Finland, 2012.

[44] T. Calinski and J. Harabasz, "A dendrite method for cluster analysis," *Communication in statistics-theory and methods,* 3(1), pp. 1–27, 1974.

[45] G. Ball and L. Hubert, "ISODATA, A novel method of data analysis and pattern classification," *Tech. Rep. Standford Research Institute Menlo Park CA,* 1965.

[46] L. Xu, "Bayesian Ying-Yang machine, clustering and number of clusters," *Pattern Recognition Letters,* 18(11), pp. 1167–1178, 1997.

[47] J. Dunn "Well separated clusters and optimal fuzzy partitions," *J. Cybernetica,* 4(1), pp. 95–104, 1974.

[48] D.L. Davies and D.W. Bouldin, "A cluster separation measure", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 1(2), pp. 95-104, 1979.

[49] P.J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. computational and applied mathematics*, *20*, pp. 53-65, 1987.

[50] X. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* 1(8), pp. 841–847, 1991.

[51] Q. Zhao and P. Fränti, "WB-index: A sum-of-squares based index for cluster validity," *Data and Knowledge Engineering*, 92, pp. 77-89, 2014.

[52] N.R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognition*, *30*(6), pp. 847-857, 1997.

[53] B.E. Dom, "An information-theoretic external cluster-validity measure," *Research Report RJ 10219*, IBM, 2001.

[54] A. Strehl, J. Ghosh, and C. Cardie, "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," *J. Machine Learning Research*, 3, pp. 583-617, 2003.

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

[56] L. I. Kuncheva, S. T. Hadjitodorov and L. P. Todorova, "Experimental comparison of cluster ensemble methods," *Int. Conf. Information Fusion*, pp. 1-7, 2006.

[57] P. Fränti, J. Kivijärvi, T. Kaukoranta, and O. Nevalainen, "Genetic algorithms for large scale clustering problems," *The Computer Journal*, 40(9), pp. 547-554, 1997.

[58] W.M. Rand, "Objective criteria for the evaluation of clustering methods," *J. American Statistical association*, *66*(336), pp. 846-850, 1971.

[59] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, pp. 193–218, 1985.

[60] T.O. Kvalseth, "Entropy and correlation: some comments," *IEEE Trans. Syst. Man Cybern.*, 17(3), pp. 517–519, 1987.

[61] J. Wu, H. Xiong and J. Chen, "Adapting the right measures for k-means clustering," *ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD'09)*, pp. 877–886, 2009.

[62] M.C.P. de Souto, A.L.V. Coelho, K. Faceli, T.C. Sakata, V. Bonadia, and I.G. Costa, "A comparison of external

clustering evaluation indices in the context of imbalanced data sets," *Brazilian Symp. Neural Networks*, pp. 49-54, 2012.

[63] M. Meila and D. Heckerman, "An experimental comparison of model based clustering methods," *Machine Learning*, 41(1-2), pp. 9–29, 2001.

[64] S.V. Dongen, "Performance criteria for graph clustering and Markov cluster experiments," *Technical Report INSR0012*, Centrum voor Wiskunde en Informatica, 2000.

[65] Q. Zhao and P. Fränti, "Centroid ratio for a pairwise random swap clustering algorithm," *IEEE Trans. Knowledge and Data Engineering,* 26(5), pp. 1090-1101, 2014.

[66] N.X. Vinh, J. Epps and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *J. Machine Learning Research*, 11, pp. 2837–2854, 2010.

[67] N.X. Vinh, J. Epps and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," *Int. Conf. Machine Learning (ICML'09)*, pp. 1073-1080, 2009.

[68] S. Wagner and D. Wagner, "Comparing clusterings – an overview," *Technical Report, 2006-4*, Fakultät für Informatik, Universit¨at Karlsruhe (TH) , 2006.

[69] M. Meila, "Comparing clusterings – an information based distance," *J. Multivariate Analysis*, 98(5), pp. 873-895, 2007.

[70] S. Choi, S. Cha and C. Tappert, "A survey of binary similarity and distance measures," *J. Systemics, Cybernetics and Informatics* 8(1), pp. 43-48, 2010.

[71] S.B. Dalirsefat, A. Meyer and SZ. Mirhoseini, "Comparison of similarity coefficients used for cluster analysis with amplified fragment length polymorphism markers in the silkworm, Bombyx mori," *J. Insect Science*, 9(1), pp. 681-689, 2009.

[72] B. Sarker, "The resemblance coefficients in group technology: A survey and comparative study of relational metrics," *Computers and Industrial Engineering*, 30(1), pp. 103–116, 1996.

[73] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," *Genome biology*, 3(7), research0036, 2002.

[74] E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity," *Neural computation* 13(11), pp. 2573-2593, 2001.

[75] T. Lange, V. Roth, M. Braun, and J. Buhmann. "Stability-based validation of clustering solutions," *Neural computation* 16(6), pp. 1299-1323, 2004.

[76] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," *Pacific symposium on biocomputing*, 7, pp. 6-17, 2001.

[77] J. N. Breckenridge, "Replicating cluster analysis: Method, consistency and validity," *Multivariate Behavioral research*, 24(2), pp.147-161, 1989.

[78] P. Smyth, "Clustering Using Monte Carlo Cross-Validation," *Int. Conf. Knowledge Discovery and Data Mining*, pp. 126-133, 1996.

[79] J. E. Overall and K. N. Magee, "Replication as a rule for determining the number of clusters in hierarchical cluster analysis," *Applied Psychological Measurement*, 16(2), pp. 119-128, 1992.

[80] J. Fridlyand and S. Dudoit, "Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method," *Tech. Report 600, Department of Statistics, UC Berkeley*, 31, 2001.

[81] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *J. Computational and Graphical Statistics*, 14(3), pp. 511-528, 2005.

[82] S. Ben-David, U. V. Luxburg, and D. Pál, "A sober look at clustering stability," *Learning theory*, Springer Berlin Heidelberg, pp. 5-19, 2006.

[83] U. V. Luxburg, "Clustering stability: An overview," *Foundations and Trends in Machine Learning*, 2(3), pp. 235-274, 2010.

[84] U. Möller and D. Radke, "A cluster validity approach based on nearest-neighbor resampling," *18th Int. Conf. Pattern recognition*, 1, pp. 892-895, 2006.

[85] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, M. Hayward, and J. Trent., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, *406*(6795), pp. 536-540, 2000.

[86] O. Abul, A. Lo, R. Alhajj, F. Polat, and K. Barker, "Cluster validity analysis using subsampling," *IEEE Int. Conf. Systems, Man and Cybernetics,* 2, pp. 1435-1440, 2003.

[87] H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration", *Pattern Recognition*, 30 (7), 1109-1119, July 1997.

Paper P1

P. Fränti, M. Rezaei, and Q. Zhao,
"Centroid index: cluster level
similarity measure",
Pattern Recognition,
47(9), pp. 3034-3045, 2014.

Pattern Recognition 47 (2014) 3034–3045

Contents lists available at ScienceDirect

Pattern Recognition

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

Centroid index: Cluster level similarity measure

Pasi Fränti [a,*], Mohammad Rezaei [a], Qinpei Zhao [b]

[a] Speech & Image Processing Unit, Department of Computer Science, University of Eastern Finland, P.O. Box 111, FIN-80101 Joensuu, Finland
[b] School of Software Engineering, Tongji Unversity, Shanghai, China

ARTICLE INFO

Article history:
Received 8 May 2013
Received in revised form
25 November 2013
Accepted 18 March 2014
Available online 29 March 2014

Keywords:
Clustering
k-Means
External validity
Similarity measure

ABSTRACT

In clustering algorithm, one of the main challenges is to solve the global allocation of the clusters instead of just local tuning of the partition borders. Despite this, all external cluster validity indexes calculate only point-level differences of two partitions without any direct information about how similar their cluster-level structures are. In this paper, we introduce a cluster level index called centroid index. The measure is intuitive, simple to implement, fast to compute and applicable in case of model mismatch as well. To a certain extent, we expect it to generalize other clustering models beyond the centroid-based *k*-means as well.

1. Introduction

Quality of centroid-based clustering is usually evaluated by internal validity indexes, most commonly by measuring intra-cluster variance. Internal validity indexes use information intrinsic to the data to assess the quality of a clustering. These include measures such as *Dunn's index* [1], *Davies–Bouldin index* [2] and *Silhouette coefficient* [3]. For a recent survey, see [4].

External indexes can be applied to compare the clustering against another solution or ground truth (if available). The ground truth can be a small representative training set given by an expert of the application domain. However, synthetic data is often used to test different aspects of the clustering methods, where their ground truth is easier to obtain. The indexes can also be applied in clustering ensemble [5,6] and used in genetic algorithms [7] to measure genetic diversity in a population. In [8], external indexes have been used for comparing the results of multiple runs to study the stability of the *k*-means algorithm. In [9], a framework for evaluating popular internal validity indexes was introduced by using external indexes on ground-truth labels. To sum up, in all these cases the main goal is to measure the similarity of two given clusterings.

Most external indexes are based on counting how many pairs of data points are co-located into the same (or different) cluster in both solutions. Examples of these are *Rand index* [10], *adjusted Rand index* [11], *Fowlkes and Mallows index* [12] and *Jaccard coefficient* [13]. A popular application-dependent approach is to measure classification error, which is quite often employed in supervised learning. Another type of external validity indexes is based on finding matches between the clusters in two solutions. Normalized *van Dongen criterion* [14,15] has a simple computation form and it can measure data with imbalanced class distributions. Other indexes utilize the entropy in different manners to compare two solutions. *Mutual information* [16] is derived from conditional entropy and *variation of information* [17] is a complement of the mutual information. Studies of external indexes can be found in [15,18].

For comparing clusterings, external indexes have been widely used by counting how many pairs of data points are partitioned consistently in the two clustering solutions. In order to be consistent, a pair of points must be allocated in both solutions either in the same cluster, or in a different cluster. This provides estimation of point-level similarity but does not give any direct information about the similarity at cluster level. For example in Fig. 1, both examples have large point-level mismatches (marked by yellow) but only the second example has cluster level mismatches.

In this paper, we propose a cluster level measure to estimate the similarity of two clustering solutions. First, nearest neighbor mapping is performed between the two sets of cluster prototypes (centroids), and the number of zero-mappings is then calculated. Each zero count means that there is no matching cluster in the other solution. The total number of zero-mappings gives direct information of how many different cluster locations are there in the two clustering solutions in total. In case of a perfect match, the index provides zero value indicating that the solutions have the same cluster-level structure. We denote the measure as *centroid index* (*CI*).

* Corresponding author.
E-mail address: franti@cs.joensuu.fi (P. Fränti).

http://dx.doi.org/10.1016/j.patcog.2014.03.017
0031-3203/

Most similar to our method are set-based measures [14,17]. They perform matching of the clusters and then measure the proportion of overlap across the matching clusters. Heuristic matching by a greedy algorithm is often done [14,31] because the optimal matching by Hungarian algorithm, for example, is not trivial to implement and takes $O(N^3)$ time. Matching problem assumes that the number of clusters is equal. If this is not the case, some clusters must be left out and dealt with another manner. The set-based methods are also restricted to measure point-level differences.

Fig. 2 demonstrates the difference between a local point-level index (Adjusted Rand index) and the new centroid index (*CI*). The results of agglomerative clustering [19,20] and random swap algorithms [21,22] have only point level differences but have the same cluster level structure. The corresponding *CI*-value is 0. The result of the *k*-means, however, has one differently allocated centroid and the corresponding *CI*-values are 1. Adjusted Rand index reflects only to point level differences (values of 0.82, 0.88 and 0.91), which have less clear interpretation in practice. The proposed index is therefore more informative.

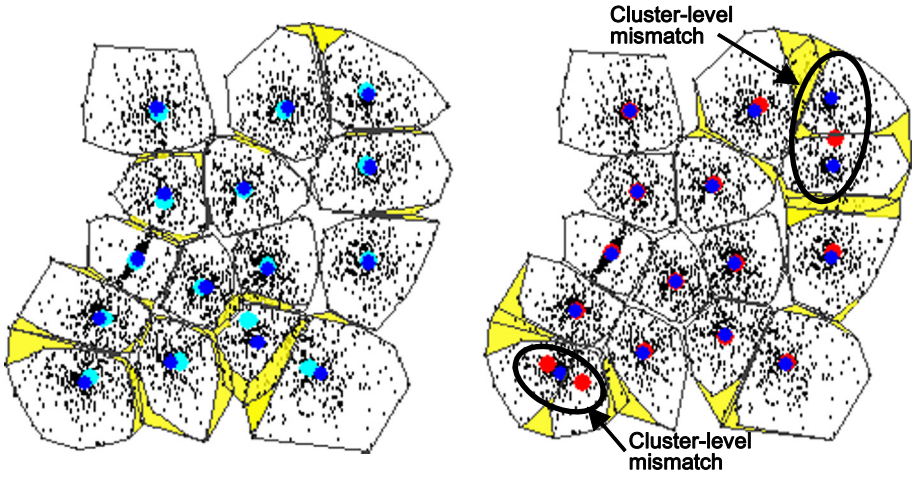


Fig. 1. Two different point-level clustering comparisons. Differences in the partitions are emphasized by yellow coloring. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

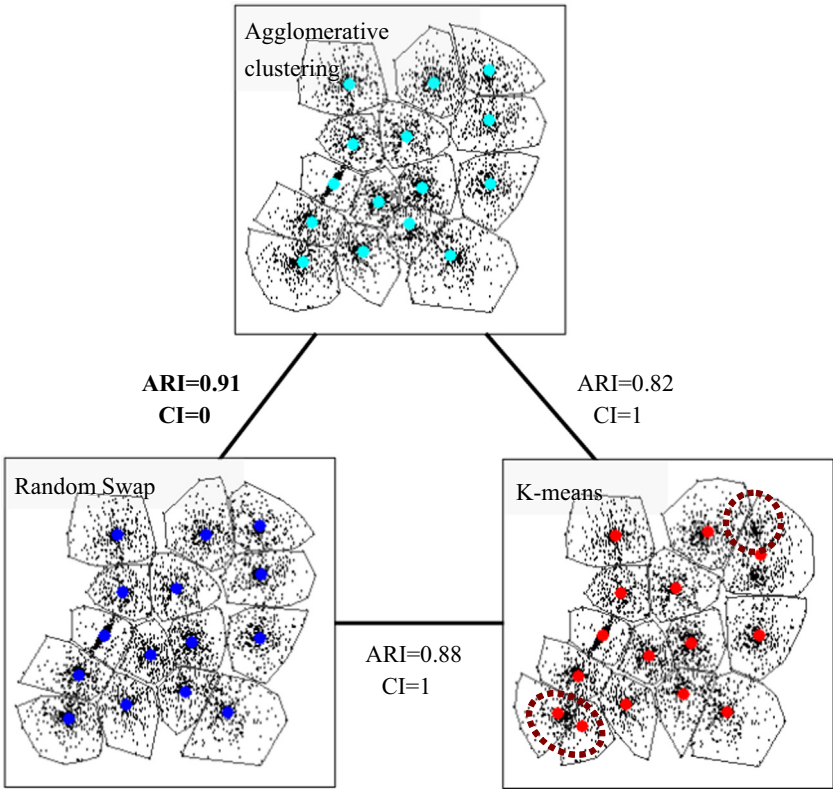


Fig. 2. Three clustering solutions and the corresponding values of Adjusted Rand index and the proposed centroid index (*CI*). The *k*-means solution has one incorrectly allocated cluster at the bottom left corner and one cluster missing at the top right corner. Otherwise the three solutions have only point level differences.

The main advantage of the centroid index is its clear intuitive interpretation. Each zero-count indicates exactly one missing cluster in the solution, either caused by different global allocation or by different number of clusters. The other benefits are that the centroid index is simple to implement and fast to compute. We expect that the main idea can be generalized to other clustering models beyond the centroid-based model (*k*-means).

The rest of the paper is organized as follows. We first define the centroid index in Section 2. We also give extension to measure point-level differences and discuss generalization to other type clustering problems. In Section 3, the index is compared against the existing indexes using artificial and real data. Furthermore, we apply the index for studying highly optimized clustering solutions and find out that it can recognize structural differences even between near-optimal clusterings that have seemingly similar partition. Another application of the index is to measure the stability of clustering algorithms. Conclusions are then drawn in Section 4.

2. Cluster level similarity

K-means clustering problem is defined as follows. Given a set of N data points x in D-dimensional space, partition the points into K clusters so that intra cluster variance (mean square error) is minimized. Centroids c_k represents the prototypes in k-means. The cost function is defined as

$$f = \frac{1}{N}\sum_{i=1}^{N}\sum_{x_i \in c_k} ||x_i - c_k||^2 \tag{1}$$

2.1. Duality of centroids and partition

Partition and the set of centroids are defined as

$$p_i \leftarrow \arg\min_{1 \le j \le M} ||x_i - c_j||^2 \quad \forall i \in [1, N] \tag{2}$$

$$c_j \leftarrow \sum_{p_i = j} x_i \;/\; \sum_{p_i = j} 1 \quad \forall j \in [1, K] \tag{3}$$

For a given partition $\{p_i\}$, the optimal prototype of a cluster is its centroid (arithmetic mean). And vice versa, for a given prototypes, optimal partition can be solved by assigning each point to the cluster whose prototype c_j is nearest. Thus, partition and centroids can be considered as *dual* structures (see also Appendix A): if one of them is given, the other one can always be uniquely determined using (2) and (3).

The duality is utilized in the *k-means algorithm* [23], which finds the nearest local minimum for a given initial solution by repeating these two properties in turn. The steps are called *partition step* and *centroid step*. However, k-means is limited to make local point-level changes only. More advanced algorithms, on the other hand, focus on solving the cluster location globally by operating with the prototypes, and solve the partition trivially by Eq. (2). Most common approach is to use k-means for the point-level fine-tuning, integrated either directly within the algorithm, or applying it as a separate post processing step.

Incremental algorithms add new clusters step by step by splitting an existing cluster [24,25], or by adding a new prototype [26], which attracts points from neighbor clusters. The opposite approach is to assign every data point into its own cluster, and then stepwise merge two clusters [27] or remove an existing one [28]. Fine-tuning can be done by k-means either after each operation, or after the entire process. Most successful iterative algorithms swap the prototypes randomly [21,22] or by deterministic manner [29], whereas genetic algorithms combine two entire clustering solutions by a crossover [30]. The success of all these algorithms is based on making cluster level changes. It is therefore reasonable that the similarity of solutions is measured at cluster level also.

2.2. Centroid index

Centroid Index (*CI*) measures cluster-level differences of two solutions. Since most essential cluster-level information is captured by the prototypes (cluster centroids), the calculations are based on them. Given two sets of prototypes $C=\{c_1, c_2, c_3, \ldots, c_{K1}\}$ and $C'=\{c'_1, c'_2, c'_3, \ldots, c'_{K2}\}$, we construct nearest neighbor mappings ($C \rightarrow C'$) as follows:

$$q_i \leftarrow \arg\min_{1 \le j \le K2} \| c_i - c'_j \|^2 \quad \forall i \in [1, K1] \tag{4}$$

For each target prototype c'_j, we analyze how many prototypes c_i consider it as the nearest ($q_i=j$). In specific, we are interested in the ones which no prototype is mapped to

$$orphan(c'_j) = \begin{cases} 1 & q_i \neq j \ \forall i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The dissimilarity of C in respect to C' is the number of orphan prototypes

$$CI_1(C, C') = \sum_{j=1}^{K2} orphan(c'_j) \tag{6}$$

We define that two clusterings (with same number of clusters $K1=K2$) have the same cluster-level structure if every prototype is mapped exactly once ($CI_1=0$). Otherwise, every orphan indicates that there is a cluster in C' that is missing in C. For example, in Fig. 3 there are two sets of prototypes. Two prototypes are orphans, which is interpreted that there are two differently allocated prototypes with respect to the reference solution.

Note that the mapping is not symmetric ($C \rightarrow C' \neq C' \rightarrow C$). Symmetric version of the index is obtained by making the mapping in both ways

$$CI_2(C, C') = \max\{CI_1(C, C'), CI_1(C', C)\} \tag{7}$$

The index has clear intuitive interpretation: it measures how many clusters are differently located in the two solutions. In specific, if there are no orphans (each prototype has been mapped exactly once in both ways), the two clustering structures are equal. This kind of bijective 1:1 mapping happens only if the solutions have the same number of clusters, and the prototypes have the same global allocation. From algorithm point of view, the value of the index indicates how many prototype need to be swapped in order to transform one of the clustering solution to the other.

2.3. Generalizations

2.3.1. Different number of clusters

With the symmetric variant (CI_2), the number of clusters does not matter because the index is not limited by the pairing as other set-based measures. Instead, it gives a value that equals to the difference in the number of clusters ($K2-K1$), or higher if other cluster-level mismatches are also detected. Intuitive interpretation of the value is the same as in Section 2.2. If the one-way variant (CI_1) is used, it should be calculated by mapping from the solution with fewer clusters to the solution with more clusters. Sample values are shown in Table 1, where three clusters found by k-means are compared to the ground truth (GT) with two clusters.

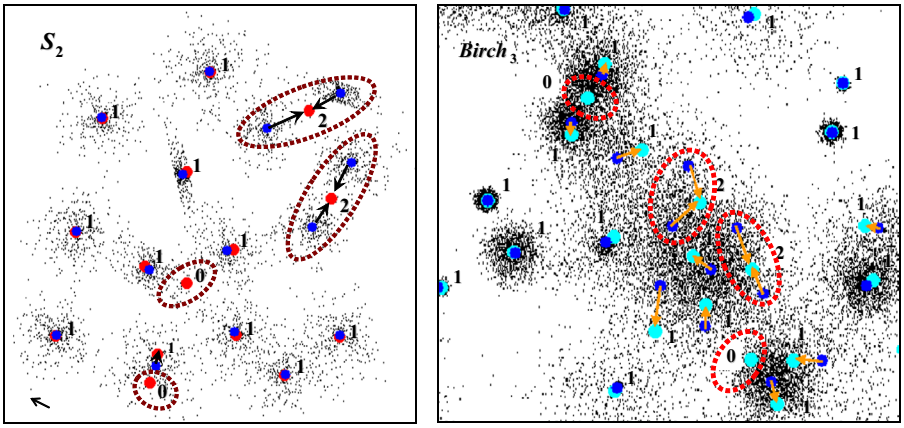


Fig. 3. Two sets of prototypes and their mappings are shown for S_2 (left) and for $Birch_3$ (right). In both examples, there are two orphans resulting to index value of $CI=2$.

Table 1
CI, CSI, Normalized van Dongen index (NVD) and Criterion-H (CH) values between the four different *k*-means clustering (3 clusters) and ground truth (GT). Perfect match are indicated by the following values: $CI=0$, NVD=0, CH=0, CSI=1.

NVD / CI	1	2	3	4	GT
1	-	0.23	0.22	0.22	0.11
2	1	-	0.13	0.13	0.12
3	1	0	-	0.22	0.11
4	1	0	1	-	0.11
GT	1	1	1	1	-

CH / CSI	1	2	3	4	GT
1	-	0.47	0.44	0.44	0.22
2	0.53	-	0.13	0.12	0.25
3	0.56	0.87	-	0.25	0.22
4	0.56	0.87	0.65	-	0.22
GT	0.78	0.75	0.78	0.78	-

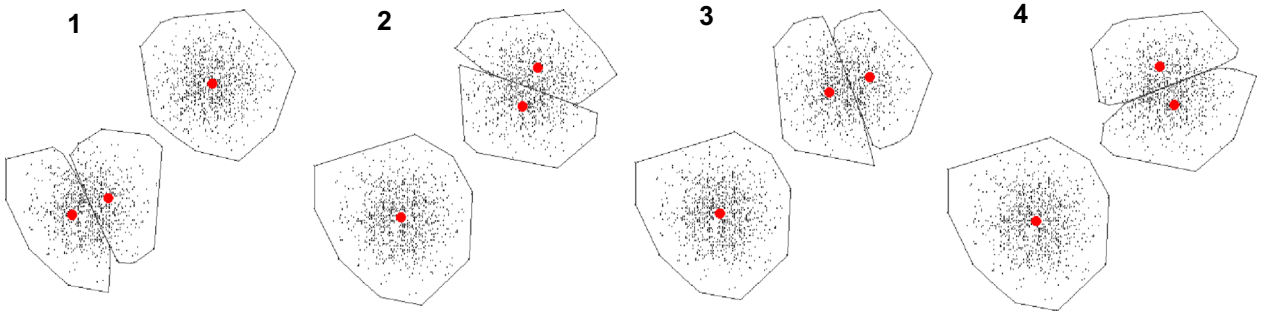


Fig. 4. Four different *k*-means solutions. Solution 1 has clearly different allocation than the others, whereas solutions 2–4 have mainly local differences.

2.3.2. *Point-level differences*

One limitation of the index is that it provides only very coarse (integer) values. This is suitable to measure cluster-level differences but not to measure more accurate point-level differences. Sample calculations are shown in Table 1 using the four sample data sets of Fig. 4. Here *CI* detects that Clustering 1 has different global allocation than 2–3–4. Among these three, the result is 0 (2–3, 2–4) or 1 (3–4) depending on the amount of variation of the topmost two clusters.

The centroid index, however, easily extends to measure point-level differences by combining it with a set-matching index [15,31] such as *criterion-H* [32] or *van Dongen index* [14]. In set-matching measures, the clusters are first paired by maximum weighted matching or by a greedy algorithm. The paired clusters are analyzed how many points they share relative to the cluster size. Our approach is simpler than that. We search for the nearest match without the pairing constraint, and allow 1:*N* type of matches. This is useful especially when the solutions have different number of clusters. Point-level centroid similarity index (CSI) can then be calculated as

$$\mathrm{CSI}=\frac{S_{12}+S_{21}}{2} \quad \text{where } S_{12}=\frac{\sum_{i=1}^{K_1} C_i \cap C_j}{N},\ S_{21}=\frac{\sum_{j=1}^{K_2} C_j \cap C_i}{N}$$

The results of CSI as well as the two set-based measures are shown in Table 1. We conclude that the point-level indexes

provide more accurate measurements than *CI* but lack the intuitive interpretation of how many clusters are differently allocated. For more thorough study of the point-level measurement and their normalizations we refer to a follow-up paper [33], which is currently under process.

For better understanding the capability and limitations of the measure, on-line visualization on 2-D data sets is available for interactive testing here: http://cs.uef.fi/sipu/clustering/animator/.

2.3.3. Other clustering models

So far we have focused on *k*-means clustering assuming that the data is in (Euclidean) vector space. This restriction, however, is not really necessary. The only requirement for the index is that we can calculate similarity between any two clusters, and in this way, find the nearest neighbor clusters in the other solution. In *k*-means, the clusters are assumed to be spherical (e.g. Gaussian) and have uniform variance, in which case the nearest neighbor is trivially found by calculating the centroid distances.

In Gaussian mixture model (GMM), each cluster (called component) is represented by the centroid and covariance matrix (often just its diagonal) in order to model elliptical clusters. In this case, it is possible to solve the nearest neighbor by finding the most similar Gaussian component as in [34]. After this, the number of orphan models can be calculated in the same way to measure the similarity of two GMMs. Potential utilization of this could be done in a swap-based EM algorithm [35].

Extension to density-based clustering is less straightforward but possible. In [36], clustering is represented by their density profiles along each attribute. Our index can be generalized using this or any other definition of the similarity between two clusters, and then performing the nearest neighbor mapping.

3. Experiments

We compare the centroid index against popular point-level external validity indexes such as adjusted Rand index (ARI) [5], normalized van Dongen (NVD) [14] and normalized mutual information (NMI) [42].

Denote the two clustering partitions by $P=\{p_1, p_2,\ldots, p_{K1}\}$ and $S=\{s_1, s_2,\ldots, s_{K2}\}$ whose similarity we want to measure. For every pair of data points (x_i, x_j), the following counts are calculated:

a = the number of point pairs in the same cluster in P and in S.
b = the number of point pairs in the same cluster in P but in different in S.
c = the number of point pairs in the same cluster in S but in different in P.
d = the number of point pairs in different clusters in P and in S.

A contingency table of P and S is a matrix with n_{ij}, which is the number of objects that are both in clusters P_i and S_j, i.e., $n_{ij}=|P_i \cap G_j|$. The pair counting index ARI is based on counting the pairs of points on which the two clusterings agree or disagree. The indexes are defined based on the contingency table as follows:

$$\mathrm{ARI}=\frac{a-(a+c)(a+b)/d}{(a+c)+(a+b)/2-(a+c)(a+b)/d} \tag{8}$$

$$\mathrm{NVD}=\frac{\left(2N-\sum_{i=1}^{K}\max_{j=1}^{K'}n_{ij}-\sum_{j=1}^{K'}\max_{i=1}^{K}n_{ij}\right)}{2N} \tag{9}$$

$$\mathrm{NMI}=\frac{MI(P,G)}{(H(P)+H(G))/2} \tag{10}$$

where $H(P)$ is the entropy of clustering P. The value indicating complete match is 0 for *NVD*, and 1 for ARI and NMI.

3.1. Data sets

We consider the data sets summarized in Table 2 consisting of four generated data sets (Fig. 5), three image data sets (Fig. 6), and *Birch* data sets [37] (Fig. 7). The points in the first set (*Bridge*) are 4×4 non-overlapping vectors taken from a gray-scale image, and in the second set (*Miss America*) 4×4 difference blocks of two subsequent frames in video sequence. The third data set (*House*) consists of color values of the *RGB* image. *Europe* consists of differential coordinates from a large vector map. The number of clusters in these is fixed to $M=256$. The data sets S_1–S_4 are two-dimensional artificially generated data sets with varying complexity in terms of spatial data distributions with $M=15$ predefined clusters.

3.2. Clustering algorithms

For generating clustering, we consider the following algorithms: *k-means* (KM), *repeated k-means* (RKM), *k-means++* [38], *X-means* [25], *agglomerative clustering* (AC) [39], *global k-means* [26], *random swap* [21], and *genetic algorithm* [30]. For more comprehensive quality comparison of different clustering algorithms, we refer to [28].

K-means++ selects the prototypes randomly one by one so that, at each selection, the data points are weighted according to their distance to the nearest existing prototype. This simple initialization strategy distributes the prototypes more evenly among the data points. Both *k*-means++ and RKM are repeated 100 times.

X-means is a heuristic hierarchical method that tentatively splits every cluster and applies local *k*-means. Splits that provide improvement according to Bayesian information criterion are accepted. Kd-tree structure is used to speed-up *k*-means.

Agglomerative clustering (AC) and Global *k*-means (GKM) are both locally optimal hierarchical methods. AC generates the clustering using a sequence of merge operations (bottom-up approach) so that at each step, the pair of clusters is merged that increases objective function value least.

Global *k*-means (GKM) uses the opposite top-down approach. At each step, it considers every data point as a potential location for a new cluster, applies *k*-means iterations (here 10 iterations) and then selects the candidate solution that decreases the objective function value most. The complexity of the method is very high and it is not able to process the largest data sets in reasonable time.

Random swap (RS) finds the solution by a sequence of *prototype swaps* and by fine-tuning their exact location by *k*-means. The prototype and its new location are selected randomly, and the new trial solution is accepted only if it improves the previous one. This iterative approach is simple to implement and it finds the correct solution if iterated long enough.

Genetic algorithm (GA) maintains a set of solutions. It generates new candidate solutions by AC-based crossover, and fine-tuned by two iterations of *k*-means. We use population of 50 candidate solutions, and generate 50 generations. In total, there are 2500 high quality candidate solutions, and the best clustering result is produced, which is also visually verified to be the global optimum (S_1–S_4, $Birch_1$, $Birch_2$).

3.3. Experiments with artificial data

We made visual comparison of the results of all algorithms against the known ground truth with all 2-D data sets. Figs. 8 and 9 show selected cross-comparison samples for S_1–S_4, $Birch_1$ and $Birch_2$. For S_1–S_4, all algorithms provide correct cluster allocation except *k*-means, *X*-means for S_2, and AC for S_4. For $Birch_1$ and $Birch_2$, AC, RS and GA all provide correct results, with $CI=0$. In all cases, it was visually confirmed that *CI* equals to the number of

Table 2
Summary of the data sets.

Data set	Type of data set	Number of data points (N)	Number of clusters (M)	Dimension of data (D)
Bridge	Gray-scale image blocks	4096	256	16
House[a]	*RGB* image	34,112	256	3
Miss America	Residual image blocks	6480	256	16
Europe	Differential coordinates	169,673	256	2
$Birch_1$–$Birch_3$	Synthetically generated	100,000	100	2
S_1–S_4	Synthetically generated	5000	15	2

[a] Duplicate data points are combined and their frequency information is stored instead.

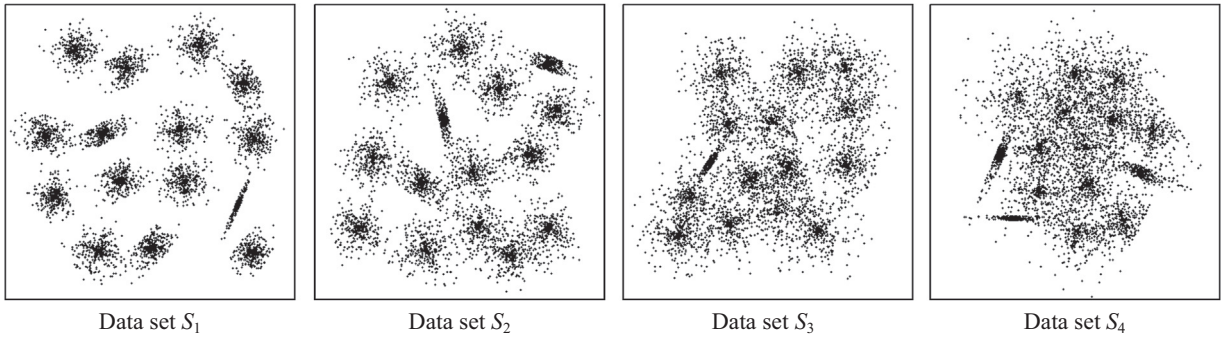


Fig. 5. Generated data sets with varying degrees of spatial overlap.

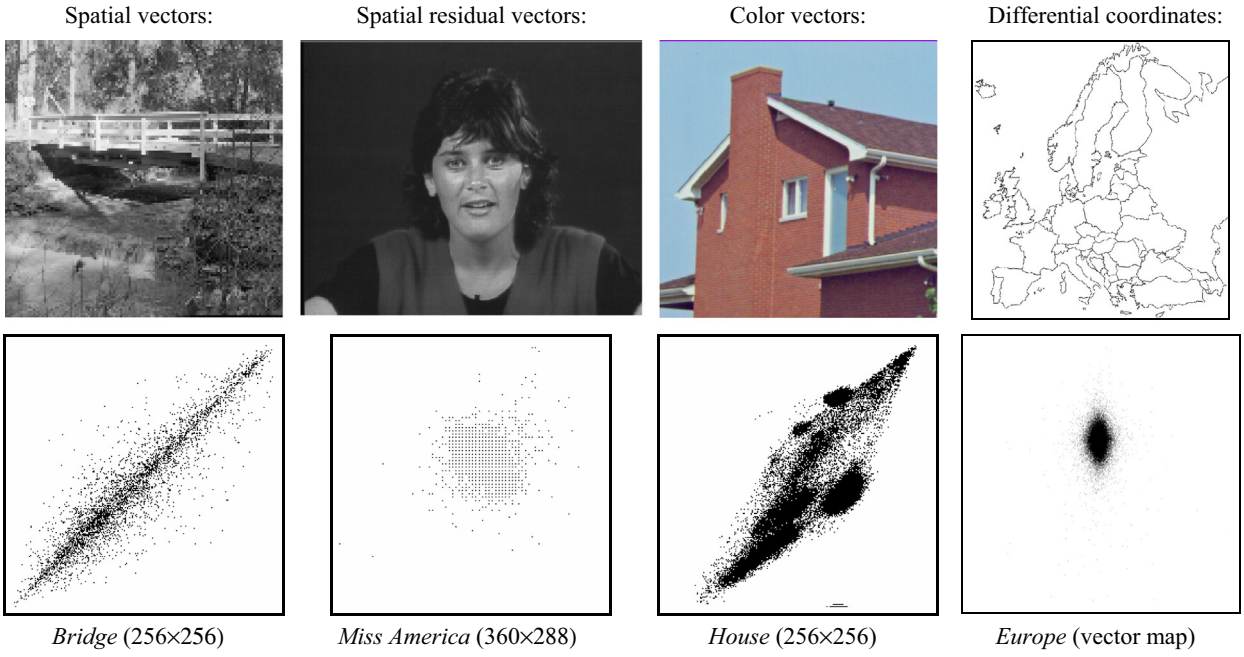


Fig. 6. Image data sets and their two-dimensional plots.

incorrectly located prototypes. Fig. 9 demonstrates the kind of clustering mistakes that typically appear.

For $Birch_3$, ground truth is not known. A visual comparison between RS and GA results is therefore provided in Fig. 10 as these algorithms provide the most similar results. Two clusters are differently located, and the other clusters have only minor point-level differences. At the lower part there are few point-level differences that demonstrate how large differences are tolerated by the *CI*-measure to be recognized as having the same cluster level structure.

3.4. Comparison of clustering algorithms

We next study the numerical results of the centroid index and the four point-level indexes. First, we report MSE values in Table 3 to give rough understanding about the clustering quality of the generated solutions. *K*-means provide clearly weaker results in all cases but it is difficult to make further conclusions about how good or bad the results are exactly. With *Bridge* we get 179.76 (KM), 173.64 (KM++), 168.92 (AC), 167.61 (RS) and 161.47 (GA) whereas the best reported value is 160.73 in [22]. With $Birch_1$, we get 5.47

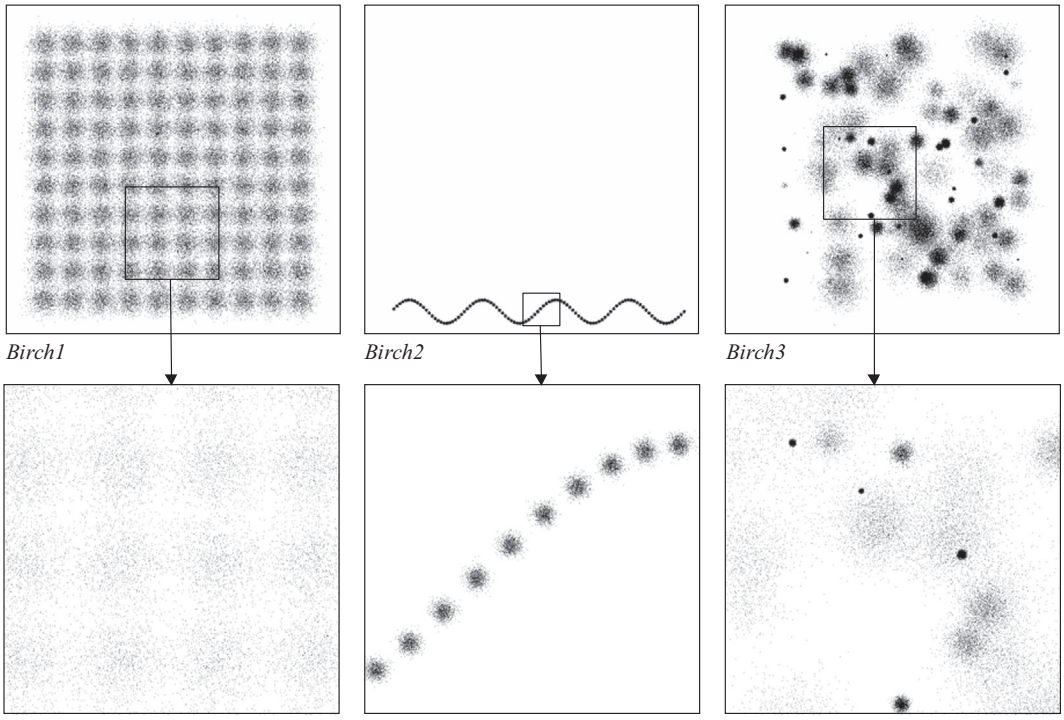


Fig. 7. *Birch* data sets.

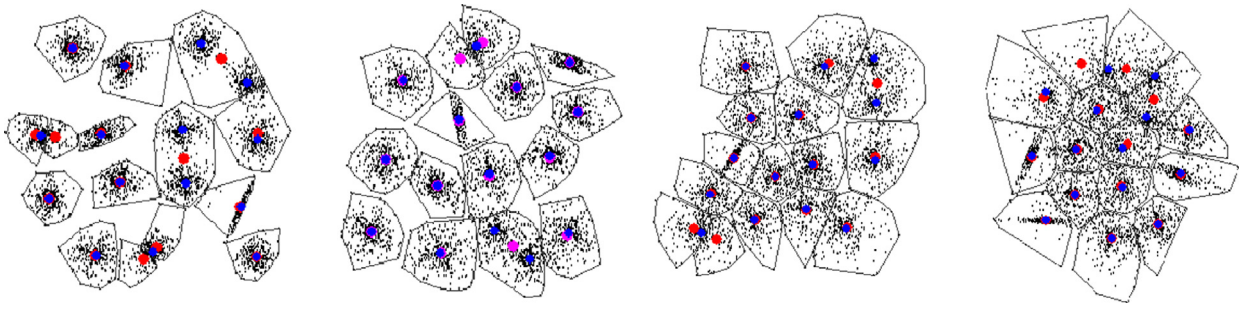


Fig. 8. Values of three indexes when comparing random swap (blue) against *k*-means (red) for S_1, S_3, S_4, and versus *X*-means (purple) for S_2. The partition borders are drawn for the *k*-means and X-means algorithms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

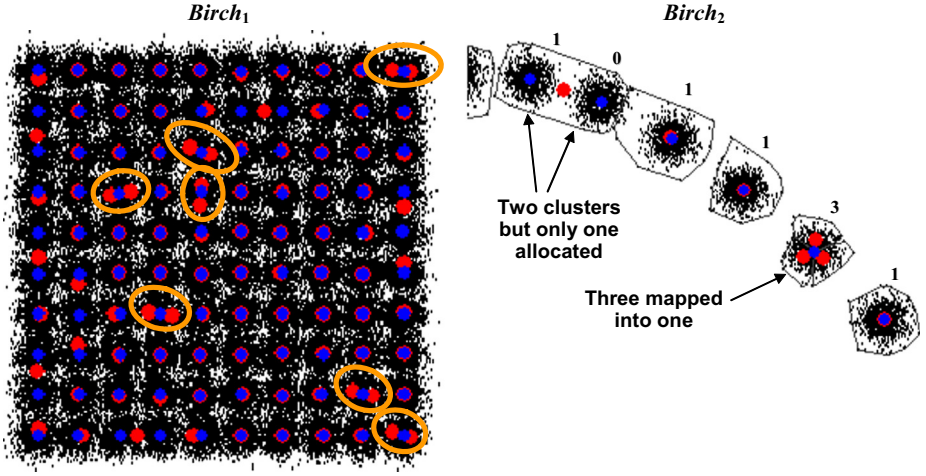


Fig. 9. *K*-means clustering (red points) versus reference solution (blue) – which is random swap clustering (left), and genetic algorithm (right). The values are $CI_2=7$ for $Birch_1$ and $CI_2=18$ for $Birch_2$ (only small fragment of the data shown here). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(KM), 4.88 (KM++), 4.73 (AC), 4.64 (RS) without any clear evidence whether the AC and KM++ results can be considered essentially similar to that of RS.

Table 4 provides the corresponding values for all the point-level indexes. Known ground truth is used as the reference solution when available (S_1–S_4, $Birch_1$, $Birch_2$) and for the remaining data sets the result of GA is used as reference.

Adjusted Rand index provides higher values for all the correct clustering results with S_1–S_4, than for any of the incorrect ones. However, the scale is highly data dependent, and there is no way to distinct between correct and incorrect clustering based on the value. The correct clustering results are measured by values 1.00 (S_1) 0.98–0.99 (S_2), 0.92–0.96 (S_3) but 0.93–0.94 (S_4). *Europe* data set is even more problematic as the measure makes almost no distinction among the clustering methods.

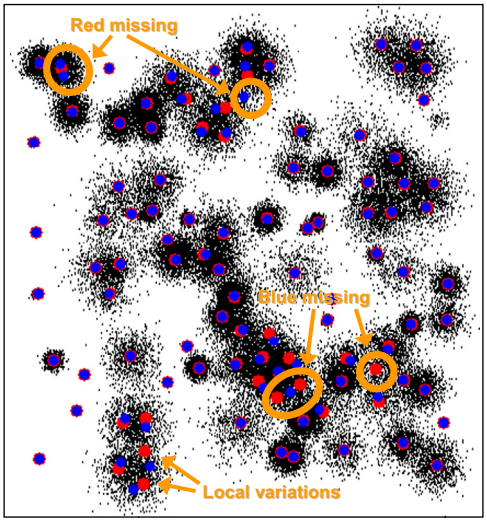


Fig. 10. Random swap (blue) versus genetic algorithm (red) with $CI_2=2$. There are two places (marked by yellow) where the results have different allocation of prototypes. In few places there are local variations of the prototypes that do not reflect to *CI*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The other two indexes perform similarly to ARI. The values of NVD are rather consistent whereas NMI provides higher variation and have the same problems with *Europe* and the S_1–S_4 sets. The point-level variant of the proposed index (CSI) provides 0.98–1.00 values when the clustering is correct. It somewhat suffers from the same problem as the other point-level indexes ($Birch_1$ for XM providing value 0.98 despite clustering is not correct) but overall it is much more consistent than ARI, NMI and NVD.

The *CI*-values are collected in Table 5. The results of S_1–S_4, $Birch_1$ and $Birch_2$ are consistent with the visual observations: the values indicate exactly how many clusters are incorrectly allocated. In specific, the index recognizes the failures of *X*-means (S_2) and AC (S_4).

With higher dimensional image sets the results cannot be visually confirmed, and since the data is not expected to have clear clusters, the interpretation is less intuitive. Nevertheless, *CI* provides good estimation of the clustering quality and is useful for comparing the algorithms. For example, we can see that agglomerative clustering (AC), random swap (RS) and Global *k*-means (GKM) provide *CI*-values varying from 18 to 42, in comparison to the values 43–75 of *k*-means. This gives more intuitive understanding how much each solution differs to that of the reference solution.

Among the algorithms, only RS, GKM and GA are capable for finding the correct cluster allocation ($CI=0$) for the data sets for which ground truth is known. Agglomerative clustering has one incorrect allocation with S_4. The improved *k*-means variants (RKM, KM++ and XM) fail to find the optimal cluster allocation for *Birch* sets, whereas the plain *k*-means fails in all cases.

3.5. Comparison of highly optimized solutions

The results in Table 5 indicate that although the best algorithms provide quite similar results in terms of minimizing the cost function (MSE), the clusters have different global allocation. For example, the results of GA (161.47) and GKM (164.78) have 33 clusters (13%) allocated differently. We therefore study whether this is an inevitable phenomenon when clustering non-trivial multi-dimensional image data.

Table 3
Clustering quality measured by internal index (variance).

Data set	Clustering quality (MSE)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	179.76	176.92	173.64	179.73	168.92	164.64	164.78	161.47
House	6.67	6.43	6.28	6.20	6.27	5.96	5.91	5.87
Miss America	5.95	5.83	5.52	5.92	5.36	5.28	5.21	5.10
Europe	3.61	3.28	2.50	3.57	2.62	2.83	-	2.44
$Birch_1$	5.47	5.01	4.88	5.12	4.73	4.64	-	4.64
$Birch_2$	7.47	5.65	3.07	6.29	2.28	2.28	-	2.28
$Birch_3$	2.51	2.07	1.92	2.07	1.96	1.86	-	1.86
S_1	19.71	8.92	8.92	8.92	8.93	8.92	8.92	8.92
S_2	20.58	13.28	13.28	15.87	13.44	13.28	13.28	13.28
S_3	19.57	16.89	16.89	16.89	17.70	16.89	16.89	16.89
S_4	17.73	15.70	15.70	15.71	17.52	15.70	15.71	15.70

Table 4
Clustering quality measured by the point-level indexes. The cases when the clustering was visually confirmed to be correct are emphasized by shading, and the six incorrect clusterings with S_1–S_4 are emphasized by **boldface**.

Data set	Adjusted Rand Index (ARI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.38	0.40	0.39	0.37	0.43	0.52	0.50	1
House	0.40	0.40	0.44	0.47	0.43	0.53	0.53	1
Miss America	0.19	0.19	0.18	0.20	0.20	0.20	0.23	1
Europe	0.46	0.49	0.52	0.46	0.49	0.49	-	1
Birch $_1$	0.85	0.93	0.98	0.91	0.96	1.00	-	1
Birch $_2$	0.81	0.86	0.95	0.86	1	1	-	1
Birch $_3$	0.74	0.82	0.87	0.82	0.86	0.91	-	1
S_1	**0.83**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.80**	0.99	0.99	**0.89**	0.98	0.99	0.99	0.99
S_3	**0.86**	0.96	0.96	0.96	0.92	0.96	0.96	0.96
S_4	**0.82**	0.93	0.93	0.94	**0.77**	0.93	0.93	0.93

Data set	Normalized Mutual Information (NMI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.77	0.78	0.78	0.77	0.80	0.83	0.82	1.00
House	0.80	0.80	0.81	0.82	0.81	0.83	0.84	1.00
Miss America	0.64	0.64	0.63	0.64	0.64	0.66	0.66	1.00
Europe	0.81	0.81	0.82	0.81	0.81	0.82	-	1.00
Birch $_1$	0.95	0.97	0.99	0.96	0.98	1.00	-	1.00
Birch $_2$	0.96	0.97	0.99	0.97	1.00	1.00	-	1.00
Birch $_3$	0.90	0.94	0.94	0.93	0.93	0.96	-	1.00
S_1	**0.93**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.90**	0.99	0.99	**0.95**	0.99	0.93	0.99	0.99
S_3	**0.92**	0.97	0.97	0.97	0.94	0.97	0.97	0.97
S_4	**0.88**	0.94	0.94	0.95	**0.85**	0.94	0.94	0.94

Data set	Normalized Van Dongen (NVD)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.45	0.42	0.43	0.46	0.38	0.32	0.33	0.00
House	0.44	0.43	0.40	0.37	0.40	0.33	0.31	0.00
Miss America	0.60	0.60	0.61	0.59	0.57	0.55	0.53	0.00
Europe	0.40	0.37	0.34	0.39	0.39	0.34	-	0.00
Birch $_1$	0.09	0.04	0.01	0.06	0.02	0.00	-	0.00
Birch $_2$	0.12	0.08	0.03	0.09	0.00	0.00	-	0.00
Birch $_3$	0.19	0.12	0.10	0.13	0.13	0.06	-	0.00
S_1	**0.09**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S_2	**0.11**	0.00	0.00	**0.06**	0.01	0.04	0.00	0.00
S_3	**0.08**	0.02	0.02	0.02	0.05	0.00	0.00	0.02
S_4	**0.11**	0.04	0.04	0.03	**0.13**	0.04	0.04	0.04

Data set	Centroid Similarity Index (CSI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.47	0.51	0.49	0.45	0.57	0.62	0.63	1.00
House	0.49	0.50	0.54	0.57	0.55	0.63	0.66	1.00
Miss America	0.32	0.32	0.32	0.33	0.38	0.40	0.42	1.00
Europe	0.54	0.57	0.63	0.54	0.57	0.62	---	1.00
Birch $_1$	0.87	0.94	0.98	0.93	0.99	1.00	---	1.00
Birch $_2$	0.76	0.84	0.94	0.83	1.00	1.00	---	1.00
Birch $_3$	0.71	0.82	0.87	0.81	0.86	0.93	---	1.00
S_1	**0.83**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.82**	1.00	1.00	**0.91**	1.00	1.00	1.00	1.00
S_3	**0.89**	0.99	0.99	0.99	0.98	0.99	0.99	0.99
S_4	**0.87**	0.98	0.98	0.99	**0.85**	0.98	0.98	0.98

Next we consider only highly optimized (near-optimal) clustering results produced by three different optimization processes:

- GAIS: Genetic Algorithm with Iterative Shrinking (long variant) [28].
- RS: Random Swap [21].
- PRS: Perturbation Random Swap (experimental algorithm).

GAIS is a variant of the genetic algorithm (GA) that uses random initial solutions and iterative shrinking as the crossover method. The best known algorithms are all based on this variant one way or another. Random Swap is another powerful optimization technique that always finds the global minimum or very close to it – if iterated long. We consider here 1.000.000 (1 M) and 8.000.000 (8 M) iterations, and an experimental alternative (PRS) that perturbs the attributes of every centroid by 2-5% after every 10 iterations.

We use three different starting points for the optimization, see Table 6. First one is a random clustering optimized by RS (RS_{8M}). The other two are different runs produced by GAIS labeled by the year when ran (GAIS-2002 and GAIS-2012). These two are further optimized by various combinations of RS and PRS aiming at the lowest possible MSE-value.

In Table 7, we compare all these high quality solutions against each other. Although their MSE-values are very close to each other, the results indicate that they all have different global allocation. In specific, the RS-optimized results have 22–25 difference cluster allocations compared to the GAIS results. However, when we compare the results within the 'GAIS-2002 family', they have exactly the same global allocation ($CI=0$). This indicates that RS is capable for optimizing the MSE further (from 160.72 to 160.43) but only via local fine-tuning while keeping the global allocation unchanged.

The same observation applies to the results of the 'GAIS 2012 family': fine-tuning by MSE is observed (from 160.68 to 160.39) but only minor (one cluster) difference in the global allocations, at most. Despite similar behavior when optimizing MSE, the two GAIS families have systematic differences in the global allocation: 13–18 differently allocated clusters, in total.

From the results we conclude that, in case of multi-dimensional image data, the index reveals existence of multiple clustering structures providing the same level of MSE-values but with different global cluster allocation. This indicates the existence of multiple global optima and that the proposed index can detect this. The point-level indexes can reveal the differences as well (into a certain extent) but without knowing the source of the differences originating from different global structure.

3.6. Stability of clustering

We next apply the index for measuring stability of clustering [40]. For this purpose, we generate from each data set 10 subsets by random sub-sampling, each of size 20% (overlap allowed). Each subset is then clustered by all algorithms. We measure the similarity of the results across the subsets within the same algorithm. In case of stable clustering, we expect the global structure to be the same expect minor changes due to the randomness in the sampling.

The results (Table 8) show that no variation is observed (0%) when applying a good algorithm (RS, GKM and GA) for the data sets S_1–S_4, $Birch_1$ and $Birch_2$. These all correspond to the case when the algorithm was successful with the full data as well (see Table 5). Results for NVD can also recognize stability for S_1 and $Birch_1$ only but not for S_2–S_4 and $Birch_2$. In general, instability can originate from several different reasons: applying inferior

Table 5
Clustering quality measured by the proposed centroid index (CI_2).

Data set	C-Index (CI_2)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	74	63	58	81	33	33	35	0
House	56	45	40	37	31	22	20	0
Miss America	88	91	67	88	38	43	36	0
Europe	43	39	22	47	26	23	---	0
Birch $_1$	7	3	1	4	0	0	---	0
Birch $_2$	18	11	4	12	0	0	---	0
Birch $_3$	23	11	7	10	7	2	---	0
S_1	**2**	0	0	0	0	0	0	0
S_2	**2**	0	0	**1**	0	0	0	0
S_3	**1**	0	0	0	0	0	0	0
S_4	**1**	0	0	0	**1**	0	0	0

Table 6
Highly optimized clustering results for *Bridge*. First three rows are reference results from previous experiments. The numbers in the parentheses refer to the number of random swap iterations applied.

	Method	MSE
GKM	Global K-means	164.78
RS	Random swap (5k)	164.64
GA	Genetic algorithm	161.47
RS_{8M}	Random swap (8M)	161.02
GAIS-2002	GAIS	160.72
+ RS_{1M}	GAIS + RS (1M)	160.49
+ RS_{8M}	GAIS + RS (8M)	160.43
GAIS-2012	GAIS	160.68
+ RS_{1M}	GAIS + RS (1M)	160.45
+ RS_{8M}	GAIS + RS (8M)	160.39
+ PRS	GAIS + PRS	160.33
+ RS_{8M} + PRS	GAIS + RS (8M) + PRS	160.28

Table 7
CI_1-values between the highly optimized algorithms for *Bridge*.

Centroid index (CI_1)									
Main algorithm:	RS_{8M}	GAIS 2002			GAIS 2012				
+Tuning 1	×	×	RS_{1M}	RS_{8M}	×	RS_{1M}	RS_{8M}	PRS	RS_{8M}
+Tuning 2	×	×	×	×	×	×	×	×	PRS
RS_{8M}	–	19	19	19	23	24	24	23	22
GAIS (2002)	23	–	0	0	14	15	15	14	16
+RS_{1M}	23	0	–	0	14	15	15	14	13
+RS_{8M}	23	0	0	–	14	15	15	14	13
GAIS (2012)	25	17	18	18	–	1	1	1	1
+RS_{1M}	25	17	18	18	1	–	0	0	1
+RS_{8M}	25	17	18	18	1	0	–	0	1
+PRS	25	17	18	18	1	0	0	–	1
+RS_{8M}+PRS	24	17	18	18	1	1	1	1	–

algorithm (k-means variants), using too small sub-sample size relative to the number of clusters, or using wrong number of clusters ($K=14$ or $K=16$ for S_1–S_4), or using inferior validity measure.

An open question is whether the stability could be used for detecting the number of clusters. Further tests would be needed as clustering tend to be stable also when only few ($K=3$) clusters are used. Thus, an external validity index such as CI alone is not sufficient for this task. This is left as future studies.

4. Conclusions

We have introduced a cluster level similarity measure called centroid index (CI), which has clear intuitive interpretation by corresponding to the number of differently allocated clusters. Value $CI=0$ indicates that the two clustering have the same global structure, and only local point-level differences may appear. Values $CI>0$ are indications of how many clusters are differently allocated. In swap-based clustering, this equals to the number of swaps needed, and an attempt has been made in [41] for recognizing the potential swaps.

The centroid index is trivial to implement and can be computed fast in $O(K^2)$ time based on the cluster centroids only. Point-level extension (CSI) was also introduced by calculating the (proportional) number of same points between the matched clusters. This provides more accurate result at the cost of losing the intuitive interpretation of the value.

The index was demonstrated to be able to recognize structural similarity of highly optimized clustering of 16-dimensional image data. General belief is that nearest neighbor search (and clustering itself) would become meaningless when dimension increases, yet the index found out similarity of the clustering structures that was not previously known. We also used the index to measure stability of clustering under random sub-sampling. The results are promising in such extent that we expect the index to be applicable for solving the number of clusters even though not in trivial manner as such. This is a point of further studies.

The centroid index is also expected to generalize to other clustering models such as Gaussian mixture models and density-based clustering. All what would be needed is to define similarity of two clusters in order to perform the nearest neighbor mapping.

Table 8
Stability of the clustering. The values are the proportion of differently allocated centroids (calculated as *Index*/*K*), across all 10 subsets, on average. Zero value implies stability in respect to random sub-sampling.

Data set	Relative CI-values (%)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
	K=100							
Birch 1	11	8	9	3	0	0	---	0
Birch 2	23	19	11	4	0	0	---	0
Birch 3	19	15	14	9	7	4	---	5
	K=16							
S_1	19	9	13	8	5	5	5	5
S_2	16	7	14	6	5	5	5	5
S_3	15	7	11	11	5	5	5	5
S_4	15	5	12	9	5	4	4	4
	K=15							
S_1	10	5	11	0	0	0	0	0
S_2	19	5	10	5	0	0	0	0
S_3	16	6	13	5	0	0	0	0
S_4	11	4	10	11	2	0	0	0
	K=14							
S_1	17	10	15	7	4	4	4	4
S_2	16	6	13	5	2	1	1	1
S_3	16	7	9	7	2	5	5	5
S_4	10	2	8	9	5	2	2	2
	K=3							
S_1	2	5	4	4	10	8	5	5
S_2	0	0	3	4	0	0	0	0
S_3	16	0	5	17	5	0	0	0
S_4	4	0	3	1	6	0	0	0

Data set	Relative NVD-values (%)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
	K=100							
Birch 1	15	11	12	5	3	1	---	1
Birch 2	16	14	9	3	0	0	---	0
Birch 3	17	16	15	10	11	9	---	9
	K=16							
S_1	9	6	4	3	1	2	2	2
S_2	12	5	5	3	3	3	3	3
S_3	15	7	12	8	8	5	5	5
S_4	14	8	11	13	11	8	8	8
	K=15							
S_1	11	6	6	4	0	0	0	0
S_2	11	4	11	3	2	1	1	1
S_3	17	8	12	6	7	3	3	3
S_4	19	9	15	11	10	5	5	5
	K=14							
S_1	12	8	8	5	5	5	5	4
S_2	14	6	11	9	4	1	1	2
S_3	14	10	15	7	11	8	9	9
S_4	16	7	16	14	14	8	7	6
	K=3							
S_1	22	16	21	22	12	16	16	13
S_2	15	13	17	13	8	11	11	13
S_3	11	2	8	12	11	2	2	2
S_4	13	8	16	8	23	8	8	8

Conflict of interest statement

None declared.

Appendix A. Duality property

An important property of centroid-based clustering is that the distortion difference originates from the movement of the centroid to any other point depends on the size of the cluster and the distance between the centroid and the point.

Lemma 2.1. *Given a subset S of points in R^d with size n, let c be the centroid of S. Then for any $z \in R^d$, there is*

$$\sum_{x_i \in S} ||x_i - z||^2 - \sum_{x_i \in S} ||x_i - c||^2 = n||c - z||^2 \tag{A1}$$

Proof. By expanding the left side, we have

$$\begin{aligned}
&\sum_{x_i \in S} ||x_i - z||^2 - \sum_{x_i \in S} ||x_i - c||^2 \\
&= \sum_{x_i \in S} (||x_i||^2 - 2x_i z + ||z||^2) - \sum_{x_i \in S} (||x_i||^2 - 2x_i c + ||c||^2) \\
&= \sum_{x_i \in S} 2x_i c - 2x_i z + ||z||^2 - ||c||^2 \\
&= 2\sum_{x_i \in S} x_i (c - z) + \sum_{x_i \in S} ||z||^2 - \sum_{x_i \in S} ||c||^2 \\
&= 2nc(c - z) + nz^2 - nc^2 = n||c - z||^2
\end{aligned}$$

The fourth equality follows from the fact that $c = 1/n\Sigma_{x_i \in S} x_i$.

For a given partition, the optimal set of prototypes is the centroid (arithmetic mean) of the clusters. And vice versa, for a given set of prototypes, optimal partition can always be obtained by assigning each point to its nearest centroid. Thus, partition and centroids are dual structures.

Lemma 2.2. *For each iteration $t \geq 0$ in k-means, we have that*

$$f\left(\left\{p_i^{(t)}\right\}_{i=1}^N\right) \geq f\left(\left\{p_i^{(t+1)}\right\}_{i=1}^N\right) \tag{A2}$$

Proof. Define $S_j^{(t+1)} = \{x \in \{x_i\}_{p_i = j}, 1 \leq j \leq M\}$, x satisfies that $||x - c_j^{(t)}||^2 < ||x - c_h^{(t)}||^2$, where $1 \leq h \leq K, j \neq h$.

According to the definition in Eq. (1),

$$\begin{aligned}
f\left(\left\{p_i^{(t)}\right\}_{i=1}^N\right) &= \sum_{j=1}^K \left(\sum_{x \in S_j} ||x - c_j^{(t)}||^2\right) = \sum_{j=1}^K \left(\sum_{h=1}^K \sum_{x \in S_j^{(t)} \cap S_h^{(t+1)}} ||x - c_j^{(t)}||^2\right) \\
&\geq \sum_{j=1}^K \left(\sum_{h=1}^K \sum_{x \in S_j^{(t)} \cap S_h^{(t+1)}} ||x - c_h^{(t)}||^2\right) = \sum_{h=1}^K \left(\sum_{j=1}^K \sum_{x \in S_h^{(t+1)} \cap S_j^{(t)}} ||x - c_h^{(t)}||^2\right) \\
&= \sum_{h=1}^K \left(\sum_{x \in S_h^{(t+1)}} ||x - c_h^{(t)}||^2\right) \geq \sum_{h=1}^K \left(\sum_{x \in S_h^{(t+1)}} ||x - c_h^{(t+1)}||^2\right) \\
&= f(\{p_i^{(t+1)}\}_{i=1}^N)
\end{aligned}$$

The second inequality follows the Lemma 2.1. Intuitively, Lemma 2.2 indicates the duality between the centroids and partitions.

References

[1] J.C. Dunn, Well separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1974) 95–104.
[2] D.L. Davies, D.W. Bouldin, Cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 95–104.
[3] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data. An Introduction to Cluster Analysis, Wiley, New York, 1990.

[4] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern. Recognit. 46 (1) (2013) 243–256.
[5] H. Wong, S. Zhang, Y. Chen, Generalized adjusted rand indices for cluster ensemble, Pattern Recognit. 45 (6) (2012) 2214–2226.
[6] A. Lourenco, A.L. Fred, A.K. Jain, On the scalability of evidence accumulation clustering, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10), 2010, pp. 782–785.
[7] P. Fränti, J. Kivijärvi, T. Kaukoranta, O. Nevalainen, Genetic algorithms for large scale clustering problems, Comput. J. 40 (9) (1997) 547–554.
[8] L.I. Kuncheva, D.P. Vetrov, Evaluation of stability of *k*-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1798–1808.
[9] I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J.M. Pérez, J. Martín, Towards a standard methodology to evaluate internal cluster validity indices, Pattern Recognit. Lett. 32 (2011) 505–515.
[10] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (1971) 846–850.
[11] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.
[12] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.
[13] R.R. Sokal, P.H.A. Sneath, Principles of Numeric Taxonom, W.H. Freeman, San Francisco, 1963.
[14] S. van Dongen, Performance Criteria for Graph Clustering and Markov Cluster Experiments, Technical Report INSR0012, Centrum voor Wiskunde en Informatica, 2000.
[15] J. Wu, H. Xiong, J. Chen, Adapting the right measures for *k*-means clustering, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09), 2009, pp. 877–886.
[16] A. Strehl, J. Ghosh, C. Cardie, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.
[17] M. Meila, Comparing clusterings – an information based distance, J. Multivar. Anal. 98 (2007) 873–895.
[18] J. Epps, N.X. Vinh, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.
[19] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration, Pattern Recognit. 30 (7) (1997) 1109–1119.
[20] P. Fränti, O. Virmajoki, V. Hautamäki, Fast agglomerative clustering using a *k*-nearest neighbor graph, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1875–1881.
[21] P. Fränti, J. Kivijärvi, Randomised local search algorithm for the clustering problem, Pattern Anal. Appl. 3 (4) (2000) 358–369.
[22] T. Kanungo, D.M. Mount, N. Netanyahu, C. Piatko, R. Silverman, A.Y. Wu, A local search approximation algorithm for *k*-means clustering, Comput. Geom. 28 (1) (2004) 89–112.
[23] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (2010) 651–666.
[24] P. Fränti, T. Kaukoranta, O. Nevalainen, On the splitting method for vector quantization codebook generation, Opt. Eng. 36 (11) (1997) 3043–3051.
[25] D. Pelleg, A. Moore, X-means: extending *k*-means with efficient estimation of the number of clusters, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00), Stanford, CA, USA, 2000.
[26] A. Likas, N. Vlassis, J.J. Verbeek, The global *k*-means clustering algorithm, Pattern Recognit. 36 (2003) 451–461.
[27] T. Kaukoranta, P. Fränti, O. Nevalainen, Iterative split-and-merge algorithm for VQ codebook generation, Opt. Eng. 37 (10) (1998) 2726–2732.
[28] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recognit. 39 (5) (2006) 761–765.
[29] B. Fritzke, The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks, Neural Process. Lett. 5 (1) (1997) 35–45.
[30] P. Fränti, Genetic algorithm with deterministic crossover for vector quantization, Pattern Recognit. Lett. 21 (1) (2000) 61–68.
[31] H. Xiong, J. Wu, J. Chen, *K*-means clustering versus validation measures: a data-distribution perspective, IEEE Trans. Syst. Man Cybern. Part B 39 (2) (2009) 318–331.
[32] M. Meila, D. Heckerman, An experimental comparison of model based clustering methods, Mach. Learn. 41 (1/2) (2001) 9–29.
[33] M. Rezaei, Q. Zhao, P. Fränti, 2014. Set matching based external cluster validity indexes (in preparation).
[34] J.E. Harmse, Reduction of Gaussian mixture models by maximum similarity, J. Nonparametric Stat. 22 (6) (2010) 703–709.
[35] Q. Zhao, V. Hautamäki, I. Kärkkäinen, P. Fränti, Random swap EM algorithm for Gaussian mixture models, Pattern Recognit. Lett. 33 (2012) 2120–2126.
[36] E. Bae, J. Bailey, G. Dong, A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings, Data Min. Knowl. Discov. 21 (2010) 427–471.
[37] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: a new data clustering algorithm and its applications, Data Min. Knowl. Discov. 1 (2) (1997) 141–182.
[38] D. Arthur, S. Vassilvitskii, *K*-means++: the advantages of careful seeding, in: ACM-SIAM Symposium on Discrete Algorithms (SODA'07), New Orleans, LA, 2007, pp. 1027–1035.
[39] P. Fränti, T. Kaukoranta, D.-F. Shen, K.-S. Chang, Fast and memory efficient implementation of the exact PNN, IEEE Trans. Image Process. 9 (5) (2000) 773–777.
[40] S. Ben-david, D. Pál, H.U. Simon, Stability of *k*-means clustering, Conference on Computational Learning Theory (COLT), LNCS 4539, Budapest, Hungary, 2007, pp. 20–34.
[41] Q. Zhao, P. Fränti, Centroid ratio for pairwise random swap clustering algorithm, IEEE Trans. Knowl. Data Eng. (2014) (in press) http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.113.
[42] T.O. Kvalseth, Entropy and correlation: some comments, IEEE Trans. Syst. Man Cybern. 17 (3) (1987) 517–519.

Pasi Fränti received his M.Sc. and Ph.D. degrees from the University of Turku, 1991 and 1994 in Science. Since 2000, he has been a Professor of Computer Science at the University of Eastern Finland. He has published 63 journals and 140 peer review conference papers, including 12 IEEE transaction papers. His current research interests include clustering algorithms and location-based recommendation systems. He has supervised 19 Ph.D.s and is currently the head of the East Finland doctoral program in Computer Science & Engineering (ECSE).

Mohammad Rezaei received his B.Sc. degree in Electronic engineering in 1996 and his M.Sc. degree in biomedical engineering in 2003 both from Amirkabir university of Technology, Tehran, Iran. Currently he is a Ph.D. student in university of Eastern Finland. His research interests include data clustering, multimedia processing, classification and retrieval.

Qinpei Zhao received the M.Sc. degrees in pattern recognition and image processing from the Shanghai Jiaotong University, China in 2007, and Ph.D. degree in computer science from the University of Eastern Finland, 2012. Her research interests include clustering, data mining, location-based applications and pattern recognition. Since 2013 she is with the Tongji University, Shanghai, China.

Paper P2

M. Rezaei and P. Fränti,
"Set matching measures for
external cluster validity",
IEEE Transactions on Knowledge and Data Engineering,
(Accepted), 2016.

Set Matching Measures for External Cluster Validity

Mohammad Rezaei, Pasi Fränti, *Senior Member, IEEE*

Abstract— Comparing two clustering results of a data set is a challenging task in cluster analysis. Many external validity measures have been proposed in the literature. A good measure should be invariant to the changes of data size, cluster size and number of clusters. We give an overview of existing set matching indexes and analyze their properties. Set matching measures are based on matching clusters from two clusterings. We analyze the measures in three parts: 1. cluster similarity 2. matching 3. overall measurement. Correction for chance is also investigated and we prove that normalized mutual information and variation of information are intrinsically corrected. We propose a new scheme of experiments based on synthetic data for evaluation of an external validity index. Accordingly, popular external indexes are evaluated and compared when applied to clusterings of different data size, cluster size and number of clusters. The experiments show that set matching measures are clearly better than the other tested. Based on the analytical comparisons, we introduce a new index called Pair Sets Index (PSI).

Index Terms— Clustering, External validity index, Cluster validation, Comparing clusterings, Normalization, correction for chance, adjustment for chance

1 Introduction

As a basic tool, *clustering* or cluster analysis partitions a set of unlabeled data objects into meaningful groups. A huge number of clustering techniques have been developed in different application fields [1]. Different algorithms or even one algorithm with different parameters can result in different partitions for the same data set. A question therefore arises that which partition best fits with the data set. Cluster validity indexes have been commonly used to address this problem [2], [3], [4], [5], [6], [7], [8], [9]. They are classified into *internal* and *external indexes* of which the former are based on information intrinsic to data while the latter measure the similarity between two clustering results of one data set. We focus on external validity indexes in this paper.

External validity indexes are used actively in searching for good clustering solutions, for example in ensemble clustering [10], [11], [12], [13], where the goal is to aggregate a set of clustering partitions. They have been used in genetic algorithms [14] to measure genetic diversity in a population. In [11], external indexes are used for comparing the results of multiple runs to study the stability of k-means. To evaluate internal validity indexes, a framework is introduced in [15] by using external indexes on ground-truth partition. Using these indexes we can identify those algorithms that generate similar partitions irrespective of data [1]. The indexes can also be used for determining the number of clusters for a data set [16], [17], [18].

External validity indexes measure how well the results of a clustering match the ground truth (if available) or another clustering [19], [20]. Several external validation measures have been studied in [7], [8], [9], [19], [20], [21], [22]. They can be categorized into *pair-counting*, *information theoretic* and *set matching* measures.

Pair-counting measures include *rand index*, *adjusted rand index*, *Jaccard coefficient, Fowlkes-Mallows index and several others* [9], [23]. They are based on counting the pairs of objects in the data set on which two different partitions agree or disagree. For instance, if two objects in one cluster in the first partition place also in the same cluster in the second partition, it is considered as an agreement. Most of the existing external validity indexes are classified in this group.

Information theoretic indexes such as *entropy*, *Mutual Information* and *variation of information* have also been used in comparing clusterings [9], [24], [25]. Mutual information measures the information that two clusterings share. Since there is no upper bound for mutual information, normalization is needed for easier interpretation and comparison [10]. A systematic study of this group of indexes, including several existing popular measures and recently proposed ones has been performed in [9].

Set matching indexes such as *F measure* [26], *criterion H* [27] and *Van Dongen* [28] are based on pairing similar clusters in two partitions. According to [24], existing indexes in this group suffer from the problem that clusters having no pair are not involved in comparison. The unmatched part of two paired clusters is also not taken into account. Taking use of the tight connection between partitions and centroids, cluster-level similarity indexes such as *Centroid Index* [20] and *Centroid Ratio* [29] employ the representatives of the clusters instead of point-level partitions. However, cluster-level indexes lack point-level information.

- *M. Rezaei is with the School of Computing, University of Eastern Finland, E-mail: rezaei@ cs.uef.fi.*
- *P. Fränti is with the School of Computing, University of Eastern Finland, FI 80110, E-mail: franti@cs.uef.fi.*

Comparison of different external validity indexes regarding to their properties have been reported in [7], [8], [9], [21], [24], [26]. Normalization and correction for chance, as desirable properties, keep the range of an index fixed in [-1, 1] or [0, 1] and make the index values comparable across different data sets. More specifically, correction for chance adjusts the index for randomness by transforming its expected value to zero. The importance of index normalization on data with imbalanced cluster distribution is discussed in [7], [26]. It is shown that the values of normalized measures are more spread in [0, 1], and have a wider range than unnormalized ones. According to [9] and [21], correction for chance is preferable when the number of data points is relatively small compared with the number of clusters. Other properties include sensitivity of an index to data size, cluster size imbalance and number of clusters. The effect of cluster size imbalance on a range of external validity indexes is analyzed in [26] and it is shown that normalization should be applied. Otherwise, an index is mostly affected by big clusters and does not detect changes in small clusters. Metric properties have been also discussed for external validity indexes and several researchers prefer metric because of the theoretical properties that exist on metric spaces [9], [21], [22], [24].

In this paper, we study set matching validity indexes by introducing and analyzing three components of the indexes: cluster similarity, matching and overall measurement. We also investigate correction for chance and show that normalized mutual information, variation of information and their adjusted forms are equivalent. We propose a new similarity index called Pair Sets Index (PSI) according to careful analysis and comparisons. Simplified form of PSI is also shown to be metric. Another contribution of the paper is to propose a new way of experiments for evaluating external indexes. The behavior of an index in comparison of clusterings with cluster size imbalance, different data size and number of clusters is extracted and analyzed systematically. We show by these experiments that set matching indexes clearly outperform other popular indexes.

2 Problem Definition

Given a data set $X \in R^d$ with N objects in a d-dimensional space, the problem of clustering is to group the data set into K clusters [14]. Given two sets of partitions $P=\{P_1, P_2,\ldots,P_K\}$ of K clusters and $G=\{G_1, G_2,\ldots,G_{K'}\}$ of K' clusters, an external validity index measures the similarity between P and G. A contingency table of P and G is a matrix where n_{ij} is the number of objects that are both in clusters P_i and G_j: $n_{ij}=|P_i \cap G_j|$, see Table 1. The sizes of clusters P_i and G_j are n_i and m_j, respectively.

An external validity index needs to satisfy several properties to be consistent and comparable for different data sets and clusterings structures.

Normalization transforms the index within a fixed range, for example [0, 1], which makes the comparison easier for data sets with different size and structure. Normalization is the most commonly agreed property in the clustering community [9]. To transform a dissimilarity index I_d to the range of [0, 1], normalization is performed as:

$$I_d^n(P,G) = \frac{I_d - \min(I_d)}{\max(I_d) - \min(I_d)} \tag{1}$$

where $min(I_d)$ and $max(I_d)$ are the minimum and maximum values of I_d.

TABLE 1
Contingency Table for Two Clustering P and G

	G_1	G_2	...	G_j	...	$G_{K'}$	Σ
P_1	n_{11}	n_{12}	...	n_{1j}	...	$n_{1K'}$	n_1
P_2	n_{21}	n_{22}	...	n_{2j}	...	$n_{2K'}$	n_2
...	...	...	...	...	...	...	...
P_i	n_{i1}	n_{i2}	...	n_{ij}	...	$n_{iK'}$	n_i
...	...	...	...	...	...	...	...
P_K	n_{K1}	n_{K2}	...	n_{Kj}	...	$n_{KK'}$	n_K
Σ	m_1	m_2	...	m_j	...	$m_{K'}$	N

The index values are expected to be constant when different random clusterings are compared with a ground truth [30]. A random partition is created by selecting random number of clusters of random size. The similarity between the random partition and the ground truth originates merely by chance. Take an example of rand index: the value of the index for two random partitions is not a constant, and is in a narrow range of [0.5, 1] instead of [0, 1]. By *correction for chance* or *adjustment*, the expected value of a similarity index is transformed to zero [21], [30]. Adjustment and normalization can be performed jointly as follows:

$$\text{Dissimilarity:} \quad I_d^{adj}(P,G) = \frac{I_d - \min(I_d)}{E(I_d) - \min(I_d)}$$
$$\text{Similarity:} \quad I_s^{adj}(P,G) = \frac{I_s - E(I_s)}{\max(I_s) - E(I_s)} \tag{2}$$

where the minimum of a similarity index (maximum of a dissimilarity index) is estimated by expected value $E(I_s)$.

Metric property has been also considered. Although a similarity / dissimilarity measure can be effective without being a metric [31], it is sometimes preferred. Considering dissimilarity index I_d and partitions P_1, P_2 and P_3, the metric properties require [22], [32]:

1. Non-negativity: $I_d(P_1,P_2) \geq 0$
2. Reflexivity: $I_d(P_1,P_2)=0$ if and only if $P_1=P_2$
3. Symmetry: $I_d(P_1,P_2)=I_d(P_2,P_1)$
4. Triangular inequality: $I_d(P_1,P_2)+I_d(P_2,P_3) \geq I_d(P_1,P_3)$

A similarity metric satisfies the following [32]:

1. Limited Range: $I_s(P_1,P_2) \leq I_0 < \infty$
2. Reflexivity: $I_s(P_1,P_2)= I_0$ if and only if $P_1=P_2$
3. Symmetry: $I_s(P_1,P_2)=I_s(P_2,P_1)$
4. Triangular inequality:
$$I_s(P_1,P_2) \times I_s(P_2,P_3) \leq I_s(P_1,P_3) \times (I_s(P_1,P_2)+I_s(P_2,P_3))$$

The triangular inequality for a similarity index I_s is derived according to the corresponding inequality for a dissimilarity index which is defined as c / I_s ($c>0$). However, other forms of the inequality are possible by defining other dissimilarities such as $\max(I_s)-I_s$. It is trivial to show

that if c/I_s (or $\max(I_s)-I_s$) is a dissimilarity metric, I_s is a similarity metric as well [32]. Hence, the metric properties for a similarity index can be checked for its corresponding dissimilarity.

Cluster size imbalance signifies that a data set can include clusters with big difference in their sizes. Some researchers argue that clusters with bigger sizes have more importance than smaller ones but in this paper we assume that each cluster has the same importance independent of its size. Invariance on the size of clusters is therefore another desired property of an index. Size of the data set should not affect on the index either.

An index should be independent on the number of clusters. Some indexes such as Rand Index give higher similarity for partitions with more clusters [22]. The index should also be applicable for comparing two clusterings with different number of clusters.

Monotonicity is another needed property. It states that the similarity of two clusterings monotonically decreases as their difference increases.

Once the above desired properties are met, then it ensures that the index values for different data sets are on the same scale and comparable. For instance, if an index gives 90% and 70% similarities, 90% should represent higher similarity. However, this is true only if the index is independent on data set and its clustering structure.

3 Pair-Counting and Information Theoretic Indexes

Pair-counting measures count the pairs of points on which the two clusterings agree or disagree. Four values are defined: *a* represents the number of pairs that are in the same cluster both in *P* and *G*; *b* represents the number of pairs that are in the same cluster in P but in different clusters in *G*; *c* represents the number of pairs that are in different clusters in *P* but in the same cluster in *G*; *d* represents the number of pairs that are in different clusters both in *P* and *G*. Values *a* and *d* count the agreements while *b* and *c* the disagreements. Examples of each case are illustrated in Fig. 1. The values of *a*, *b*, *c* and *d* can be calculated from the contingency table [30] as follows:

$$
\begin{aligned}
a &= \frac{1}{2}\sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}(n_{ij}-1) \\
b &= \frac{1}{2}\Big(\sum_{j=1}^{K'} m_j^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\Big) \\
c &= \frac{1}{2}\Big(\sum_{i=1}^{K} n_i^2 - \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2\Big) \\
d &= \frac{1}{2}\Big(N^2 + \sum_{i=1}^{K}\sum_{j=1}^{K'} n_{ij}^2 - \Big(\sum_{i=1}^{K} n_i^2 + \sum_{j=1}^{K'} m_j^2\Big)\Big)
\end{aligned}
\tag{3}
$$

Some of the popular indexes are listed in Table 2. *Rand index* (RI) is a well-known pair-counting measure. For random partitions, the similarity between two clusterings is desired to be close to zero. However, the expected value of rand index for random partitions is 0.5 and the index is within a narrow range of [0.5, 1] according to [11], [12], [30]. Hence, a corrected-for-chance version called *adjusted rand index* (ARI) was introduced in [30] which is upper bounded by one and lower bounded by zero. The expected value of the rand index is estimated using hyper-geometric distribution assumption in which the size and number of clusters are fixed [30].

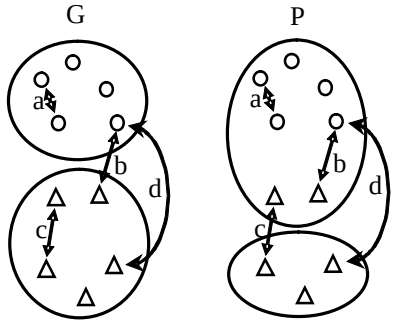


Fig. 1. The principle of pair-counting measures.

Existing information theoretic measures employ the concept of entropy [25] to compare two partitions. Entropy is measured by the average number of bits needed to store or communicate data. The entropy of clustering *P* with *K* clusters is defined as:

$$H(P) = -\sum_{i=1}^{K} p(P_i)\log p(P_i) \tag{4}$$

where $p(P_i)=n_i/N$ is the estimated probability of the cluster P_i.

Having clustering *G* and the joint distribution $p(P,G)$, the average number of bits for *P* is derived by conditional entropy [19] as follows:

$$H(P\,|\,G) = \sum_{i=1}^{K}\sum_{j=1}^{K'} p(P_i,G_j)\log p(P_i\,|\,G_j) \tag{5}$$

where the probability $p(P_i,G_j)$ can be estimated from the contingency table as n_{ij}/N.

Mutual information (MI) [9], [10] is derived from conditional entropy and represents the similarity of two clusterings [22]. If we choose a random object in the data set, knowing its cluster in *G*, mutual information measures the reduction in uncertainty of the object's cluster in *P* [22], [24]. Mutual information is defined formally as follows:

$$MI(P,G) = H(P) - H(P\,|\,G) = H(P) + H(G) - H(P,G) \tag{6}$$

In terms of probabilities, it is:

$$MI(P,G) = \sum_{i=1}^{K}\sum_{j=1}^{K'} p(P_i,G_j)\log\frac{p(P_i,G_j)}{p(P_i)p(G_j)} \tag{7}$$

Variation of Information (VI) [24] is complement of the mutual information, see Fig. 2, and is calculated by summing up the conditional entropies $H(P\,|\,G)$ and $H(G\,|\,P)$, see (8). Normalization of MI and VI is discussed in section 5.

$$
\begin{aligned}
VI(P,G) &= H(P\,|\,G) + H(G\,|\,P) = \\
&H(P) + H(G) - 2MI(P,G) = \\
&2H(P,G) - H(P) - H(G)
\end{aligned}
\tag{8}
$$

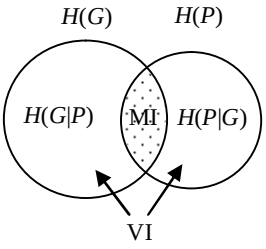


Fig. 2. Mutual information (MI) and variation of information(VI).

4 Set Matching Indexes

Set matching indexes are based on matching entire clusters. Similar clusters are first found either by pairing or matching, and their similarity is then measured using set matching methods. We classify the set matching indexes into two types: point-level and cluster-level.

Point-level indexes consider the intersection of paired clusters in two clusterings. Purity is an example of this group and it assumes one of the clusterings as ground truth [33]. Accuracy defined in [34] is equivalent (exactly the same) to Purity. Some authors use terms such as classification accuracy [35] or classification error [9] with refereeing to accuracy in [34] but this is not correct because they have other definitions in classification problem. F measure (FM) [26], Criterion H (CH) [27] and normalized Van Dongen (NVD) [28] are other set matching measures.

Cluster-level indexes include Centroid Index (CI) [20] and Centroid Ratio (CR) [29]. They use only cluster prototypes in contrast to point-level indexes which employ the labels of all objects in resulting partitions. Cluster level indexes are fast to calculate [20], and they provide clear interpretation about the differences in cluster-level structure. For example, CI=1, demonstrates one difference in the global allocation of the two clusterings. However, they do not measure partial cluster differences. Centroid Similarity Index (CSI) was introduced in [20] to extend CI to a point-level measure.

Set matching measures involve three design questions:

1. How to match the clusters
2. How to measure the similarity of two clusters
3. How to calculate overall similarity

Normalization and correction for chance (if applied) are also essential parts of the overall similarity derivation. We next give a detailed analysis of all these questions including the normalization.

1. *Similarity of two clusters*

Let P_i and G_j be two clusters in P and G respectively. Most of the set matching measures use $|P_i \cap G_j|$ to calculate the similarity of the two sets. For example, in Fig. 5, clusters G_1 and P_1 are more similar than G_2 and P_2 since the number of shared objects is 6 and 4 respectively. CH, NVD, CSI and Purity use this measure. Many other ways to measure similarity of two sets exist in literature and any of them can be employed for calculating the similarity of two clusters. Among the 76 methods listed in [36], we mention three popular ones: Jaccard [37], Sorensen-Dice [38] and Braun-Banquet [36].

TABLE 2
External Validity Indexes

Pair-counting measures	
Rand index [30]	$RI = \frac{a+d}{N(N-1)/2}$
Adjusted rand index [30]	$ARI = \frac{RI - E(RI)}{1 - E(RI)}$
Information theoretic measures	
Mutual information [25]	$MI = \sum_{i=1}^{K}\sum_{j=1}^{K'} \frac{n_{ij}}{N} \log \frac{N n_{ij}}{n_i m_j}$
Normalized Mutual Information type 1 [25]	$NMI = \frac{MI(P,G)}{(H(P)+H(G))/2}$
Normalized Mutual Information type 2 [25]	$NMI = \frac{MI(P,G)}{\sqrt{H(P)\times H(G)}}$
Normalized Variation of Information [7]	$NVI = \frac{H(P)+H(G)-2MI(P,G)}{H(P)+H(G)}$
Set matching measures	
F measure [26]	$FM = \frac{1}{N}\sum_{i=1}^{K} n_i \max_j \frac{2n_{ij}}{n_i + m_j}$
Criterion H [27]	$H = 1 - \frac{1}{N}\max_j \sum_{i=1}^{K} n_{ij}$
Normalized Van Dongen [28]	$NVD = \frac{2N - \sum_{i=1}^{K}\max_{j=1}^{K'} n_{ij} - \sum_{j=1}^{K'}\max_{i=1}^{K} n_{ij}}{2N}$
Purity [33]	$Purity = \frac{1}{N}\sum_{i=1}^{K} \max_{\pi} n_{i,\pi}(i)$
CI [20]	$CI_1(P,G) = \sum_{i=1}^{K'} orphan(G_i)$ $CI_2(P,G) = \max(CI_1(P,G), CI_1(G,P))$
CSI [20]	$CSI = \frac{\sum_{i=1}^{K} n_{ij} + \sum_{j=1}^{K'} n_{ji}}{2N}$ i, j: indexes of matched clusters
CR [29]	$CR = 1 - \sum_{i=1}^{K} \gamma_i / K$ $\gamma_i = \begin{cases} 1 & unstable\ pair \\ 0 & stable\ pair \end{cases}$
PSI	$\begin{cases} \frac{S - E(S)}{\max(K,K') - E(S)} & S \geq E(S), \max(K,K') > 1 \\ 0 & S < E(S) \\ 1 & K = K' = 1 \end{cases}$ $S = \sum_{i=1}^{\min(K,K')} \frac{n_{ij}}{\max(n_i, m_j)}$ i, j: indexes of paired clusters

$$J = \frac{|P_i \cap G_j|}{|P_i \cup G_j|} \quad (9)$$

$$SD = \frac{2|P_i \cap G_j|}{|P_i| + |G_j|} \quad (10)$$

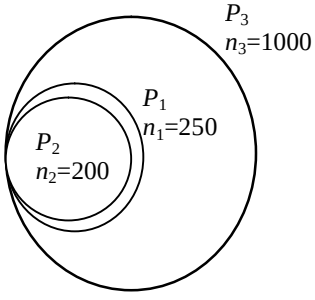
$$BB = \frac{|P_i \cap G_j|}{\max(|P_i|, |G_j|)} \quad (11)$$

These measures are in the range of [0, 1]. Distance forms of J and SD are defined as (1-J) and (1-SD) where the former is a true metric but the latter does not satisfy triangular inequality. In order to make the measure independent on cluster size, these measures normalize the number of shared objects $|P_i \cap G_j|$ according to the size of clusters in three different ways.

For example, consider the three clusters in Fig. 3 where we want to find out the more similar cluster to P_1 from P_2 or P_3. Similarity of P_1 and P_2 should be much higher than the similarity of P_1 and P_3 even though P_1 and P_3 share more objects. J, SD and BB give more intuitive similarity values than intersection. When comparing P_1 and P_3, the similarity 0.25 of J and BB is better than the 0.4 of SD. It is trivial to show that J≤BB≤SD for any two sets.

FM [22] uses *precision* and *recall* concepts by measuring n_{ij}/n_i and n_{ij}/m_j respectively. The criterion 2×precision×recall/(precision+recall) would be equivalent to SD but it avoids the normalization by cluster size using n_i×SD instead of SD.

Cluster-level indexes provide binary result (0 or 1), indicating whether the clusters have 1:1 match (CI), or the pair of clusters is unstable (CR). Table 3 lists the criteria for set matching indexes.



	P_1, P_2	P_1, P_3
J	0.80	0.25
SD	0.89	0.40
BB	0.80	0.25

Fig. 3. The effect of cluster size on cluster comparison

2. *Matching*

For every cluster, we need to find the pair to which the similarity is measured. Three cases are considered: 1. optimal pairing 2. greedy pairing 3. matching. Matching is performed based on nearest neighbor mapping so that any cluster in *P* is matched to a cluster in *G* with maximal similarity. Several clusters can be matched with the same cluster in the other clustering. Pairing is a special case of matching in which clusters are only allowed to be matched once. FM, NVD, Purity, CI and CSI employ matching whereas CH and CR use greedy pairing. We will use optimal pairing.

TABLE 3
CRITERIA FOR SIMILARITY OF TWO CLUSTERS

	Similarity criteria
FM	$\|P_i\|$×SD
H	$\|P_i \cap G_j\|$
NVD	$\|P_i \cap G_j\|$
Purity	$\|P_i \cap G_j\|$
PSI	BB
CI	0/1 (mapped or unmapped)
CSI	$\|P_i \cap G_j\|$
CR	0/1 (stable or unstable)

Matching results, in general, is not symmetric when finding pairs for clusters of *P* from *G* and vice versa. To make the index symmetric, the similarity results in both directions are usually combined, see NVD, CI and CSI equations in Table 2. FM and Purity assume that we compare a clustering against ground truth and they therefore consider matching in one direction only. Matching criterion in NVD and Purity is the number of shared objects; CI and CSI are based on similarity of prototypes.

Pairing problem, however, is not trivial to solve and different algorithms have been proposed to find approximate or optimal solution. The pairing can be seen as a matching problem in weighted bipartite graph where the nodes represent the clusters, see Fig. 4. Greedy pairing is mostly used with time complexity of $O(N^2)$. Two most similar clusters are iteratively matched and excluded. Instead of greedy pairing, we apply here Hungarian algorithm which finds the optimal solution with time complexity $O(N^3)$ where *N* is the maximum number of clusters in *P* and *G*.

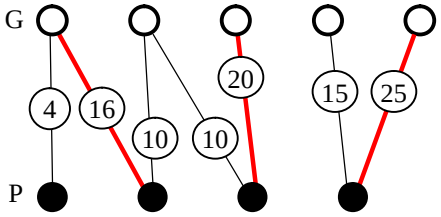


Fig. 4. Pairing clusters to maximize overall similarity. The thick lines show the optimal pairing where overall similarity according to number of shared objects would be (25+20+16) = 61.

Fig. 5 demonstrates the matching from *G* to *P* based on the number of shared objects where P_2 remains unmatched. The matching from *P* to *G* will be different and the same as greedy pairing based on number of shared objects, resulting to (P_1,G_1), (P_2,G_2) and (P_3, G_3).

Fig. 6 shows matching in CI when there is different number of clusters. We assume that the objects are in 2-*D* Euclidean space; the centroids have been shown with crosses signs. In matching *P* to *G*, one orphan centroid is produced that indicates one difference in global allocation. NVD results the same matching as CI in this example. In general, if a cluster P_i has more shared objects with G_j than G_k, the probability that its centroid is also closer to

G_j is higher. Although, this is not always true as it depends on the distribution of data among clusters. It anyway implies that the matching using intersection criterion and centroid distance are expected to produce the same result.

Fig. 7 demonstrates the results with too few (above) and too many clusters (below) compared to another with the same clustering problem or to the correct clustering. In this example, both matching and pairing are performed

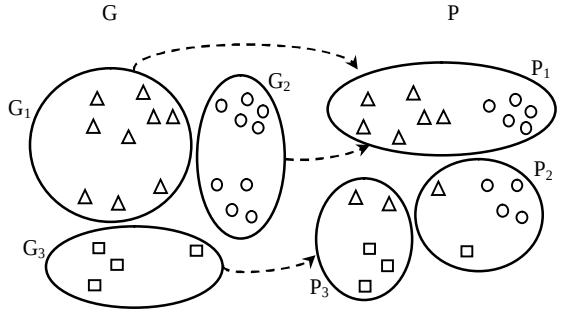


Fig. 5. Matching clusters based on maximum shared objects. Cluster P_2 remains unmatched. In pairing process of CH, G_2 is paired with P_2 after excluding G_1 and P_1 as the first pair.

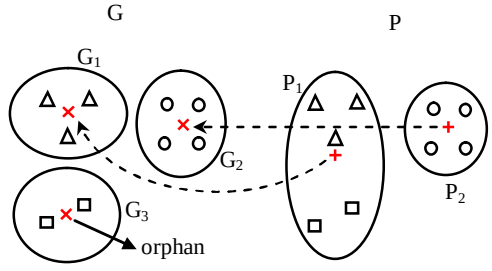


Fig. 6. Matching centroids from *P* to *G* based on nearest neighbor mapping used in CI and CSI; One orphan centroids shows one difference in global allocation.

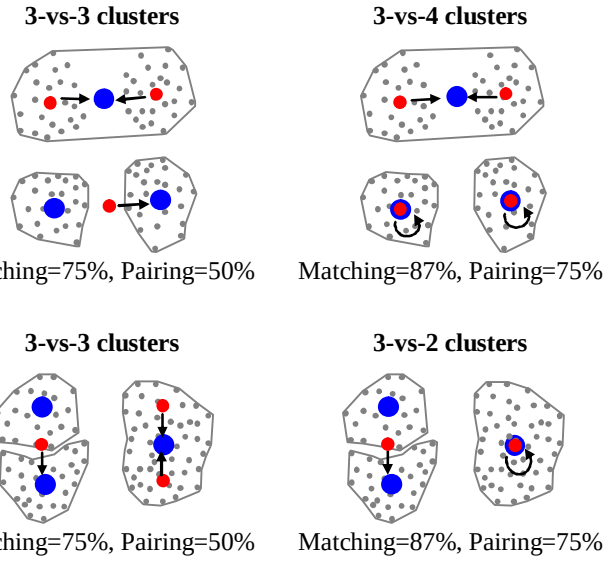


Fig. 7. Matching and pairing when too few (above) and too many (below) clusters exist. Arrows show matching from red to blue centroids. Pairing would use only part of those arrows because each cluster can be matched only once.

TABLE 4
SUMMARIZATION OF MATCHING METHODS OF INDEXES

	Pairing/ Matching	Matching criterion	Algorithm
FM	Matching	SD	One-way
CH	Pairing	$\|P_i \cap G_j\|$	Greedy
NVD	Matching	$\|P_i \cap G_j\|$	Two-way
Purity	Matching	$\|P_i \cap G_j\|$	One-way
PSI	Pairing	BB	Optimal
CI	Matching	Centroid distance	Two-way
CSI	Matching	Centroid distance	Two-way
CR	Pairing	Centroid distance	Greedy

based on number of shared objects. Matching results always higher values than pairing because in pairing some centroids remain unpaired. Pairing is more sensitive to differences in clustering structure. The result is also lower with 3-vs-3 than when comparing to the correct number of clusters (3-vs-4 and 3-vs-2). In comparing two clustering with different number of clusters, unpaired clusters indicate a disagreement on the number of clusters, which is an advantage of pairing. Table 4 summarizes the matching methods for several indexes.

3. *Overall similarity*

Overall similarity is obtained by summing up the similarities of all the matched clusters. The upper bound of overall similarity for CH is *N* (total number of objects) which is used for normalization, see Table 2. To remove the asymmetry effect of matching, NVD and CSI use 2*N* because of two-way matching, see Table 2. If we define the distance form of CSI and Purity as (1-CSI) and (1-Purity), NVD, CH, Purity and CSI are all equivalent if their matching results are the same. In fact, if matching in NVD and CSI is symmetric (*K*=*K*'), they would equal to CH and we can write:

$$NVD = 1 - \frac{\sum_{i=1}^{K} n_{ij} + \sum_{j=1}^{K'} n_{ji}}{2N} = 1 - \frac{2\sum_{i=1}^{K} n_{ij}}{2N} = 1 - \frac{\sum_{i=1}^{K} n_{ij}}{N} = CH = 1 - Purity = 1 - CSI \quad (12)$$

The overall dissimilarity of CI equals the number of zero mapped centroids of *G*. Since CI is not symmetric, CI_2 is defined as max(CI(*P*,*G*), CI(*G*,*P*)) [20]. Centroid index represents the number of differences in global allocations and it is in the range of [0, *K*-1] where *K* is the maximum number of clusters in the two clusterings. At least one non-zero mapped centroid exists and the upper bound therefore becomes *K*-1.

Centroid ratio (CR) defines the concept of (un)stable centroids. Consider a paired centroid C_i and C'_j with distance D_{ij} from clusterings *P* and *G*, respectively. Assume that the distances of C_i to the nearest centroid in *P* and C'_j to the nearest centroid in *G* are D_i and D_j. Then, if $D_{ij}^2 / (D_i \times D_j) > 1$, the pair is considered unstable. The overall similarity is defined based on the number of unstable

pairs [29], see Table 2. Table 5 summarizes the overall similarity derivation for the above mentioned indexes.

TABLE 5
OVERALL SIMILARITY DERIVATION

	Total summation	Range	Normalization
FM	similarity of matched clusters	[0, 1]	N
CH	Shared objects	[0, 1]	N
NVD	Shared objects in both directions	[0, 1]	$2N$
Purity	Shared objects in one direction	[0, 1]	N
PSI	Normalized similarity of paired clusters	[0, 1]	K
CI	Orphan clusters	[0, K-1]	-
CSI	Shared objects in both directions	[0, 1]	$2N$
CR	Unstable clusters	[0, 1]	K

5 CORRECTION FOR CHANCE

Normalization makes comparisons easier for different data sets. Correction for chance removes the similarity of two clusterings which merely originates by chance [21].

An index is normalized using its lower and upper bounds as in (1). Correction for chance can be jointly performed with normalization according to (2). Some indexes do not have fixed lower or upper bounds. For example, several upper bounds have been proposed to normalize MI [9], [25].

In comparison of two clusterings *P* and *G*, the number and size of clusters are known. To consider the effect of random partitioning, the objects of clustering *P* are distributed randomly in clusters of *G* and the expected similarity value is calculated. This is called hyper-geometric distribution assumption and was first used for deriving ARI [30].

The measures in the pair-counting class as listed in [23] are in the ranges of [0,1], [-1, 1], [0.5, 1] or [-0.25, 0.25] that further clarifies the necessity of normalization. Since all the indexes are defined based on values *a*, *b*, *c*, and *d* in (3), the upper and lower bounds are simple to derive. Many of them become equivalent after applying correction for chance [21]. ARI is the most well-known and widely used index of this group [9].

In set matching measures, the overall similarity is derived either by summing up the number of shared objects or the similarities of the matched clusters. For example, NVD, CH, Purity and CSI sum up the number of shared objects and use the total number of objects for normalization. The similarity index proposed by Larsen and Aone [39] is calculated by summing up the normalized similarities (in the range of [0, 1]) of the matched clusters. In this case, the overall similarity is normalized for each cluster individually.

Both MI and VI are metric but they are not bounded to a fixed range [22]. Mutual information of clusterings *P* and *G* is lower bounded by zero. Geometric or arithmetic mean of entropies as an upper bound can be an option for normalization (type 1 and 2 in Table 2) [22], [25], [10]. In [25] *min*(*H*(*P*), *H*(*G*)) and *max*(*H*(*P*), *H*(*G*)) are also used for normalization. An upper bound for VI is $H(P)+H(G)$, which means that clusterings *P* and *G* do not share any information [7]. The upper bound can therefore be used for normalization of VI. To derive adjusted mutual information according to (2), obtaining the expected value *E*(MI) is the key issue. An analytical formula for the expected value of mutual information is derived in [21] under the assumptions of hyper-geometric model of randomness. In [9], upper bounds for the expected value are given, and shown that, under certain assumptions, the adjusted MI measures derived based on different upper bounds become equivalent to the normalized MI measures.

We prove next that the adjusted forms of mutual information (AMI) and variation of information (AVI_s) are equivalent to their normalized forms (NMI, NVI_s) when the summation of the entropies $H(P)+H(G)$ is used for normalization.

Theorem 1. *Under hyper-geometric distribution assumption:*

$$AVI_s = NVI_s = AMI = NMI \tag{13}$$

where NVI_s and AVI_s denote the similarity form of NVI and AVI (1-NVI and 1-AVI) respectively.

Proof. See Appendix A.

6 PAIR SETS INDEX

In this section, we present a new set matching based measure called Pair Sets Index (PSI), which is designed so that the properties discussed in section 2 are all satisfied. The components of the proposed index are known but some of them are new in this context, and the overall combination is novel. In specific, PSI contains optimal pairing of the clusters (new), set matching measure using BB (new), the overall similarity measure in (14) (used also by CR), and the correction for chance (used by pair-counting and information theoretic methods only).

6.1 Similarity Measure

Given clusterings *P* and *G*, the first step is to find the pairs of clusters in two partitions. Pairing clusters in *P* and *G* is done by maximizing total similarity which is defined as:

$$S(P,G) = \sum_i S_{ij} \tag{14}$$

where S_{ij} denotes the similarity between clusters P_i and G_j and is calculated as from Braun-Banquet formula [36] as follows:

$$S_{ij} = \frac{n_{ij}}{\max(|P_i|,|G_j|)} \tag{15}$$

Here n_{ij} is the number of shared objects in the two clusters and $|P_i|$ and $|G_j|$ denote their sizes.

The corresponding distance variant is defined as $D_{ij}=1-S_{ij}$. Pairing clusters is solved as an *assignment problem* in a

bipartite graph, see Fig. 4, by minimizing the total distance. We use Hungarian algorithm to find the perfect matching in this assignment problem [40].

6.2 Correction for Chance

In this section, we describe the process of correction for chance for the proposed similarity measure in (14) and derivation of the final formula for the Pair Sets Index (PSI).

Obtaining the expected value is the key point to derive the adjusted version of the index. To derive the expected value, consider a random shuffling of P as P' under hyper-geometric distribution assumption where the number and size of the clusters in P' and P are the same. The objects of cluster G_j are distributed randomly in the clusters in P'. A larger cluster in P' gets more objects from G_j. Therefore, the number of shared objects of clusters G_j and P'_i is proportional to the size of P'_i. The number of objects of G_j (m_j) that places in P'_i (n_i) is $m_j \times (n_i / N)$, which is the number of shared objects between these two clusters when random partition P' is assumed.

Theorem 2. The maximum total similarity in (14) is achieved when the largest cluster in P' is paired with the largest one in G, and recursively the same applies to the rest of the clusters. Applying this greedy pairing, the expected value is:

$$E = \sum_{i=1}^{\min(K,K')} \frac{m_i \times (n_i / N)}{\max(m_i, n_i)} \tag{16}$$

where the size of clusters in P' is $n_1 > n_2 > \ldots > n_K$ and in G is $m_1 > m_2 > \ldots > m_{K'}$.

Proof. See Appendix B.

Next, we show that $E \leq 1$. Assuming $m_i = n_i, \forall i$, the summation in (16) is $(n_1+n_2+\ldots+n_{Kmin}) / N \leq 1$. Suppose that $n_i \geq m_i$, $\exists i$, the summation then becomes:

$$E=(n_1+n_2+\ldots+m_i+\ldots+n_{Kmin})/N$$
$$\leq (n_1+n_2+\ldots+n_{Kmin})/N \leq 1 \tag{17}$$

Therefore, it is always true that $E \leq 1$. Applying the results to (2), the adjusted index becomes:

$$PSI = \begin{cases} \dfrac{S-E}{\max(K,K')-E} & S \geq E, \max(K,K') > 1 \\ 0 & S < E \\ 1 & K = K' = 1 \end{cases} \tag{18}$$

where S is the total similarity from (14). In random partitioning S=E, PSI=0 and in a perfect match S=K, PSI=1. If there is a disagreement on the number of clusters, K≠K', max(K, K') is taken in (18) to achieve a lower similarity that reflects the disagreement. The expected value is not necessarily the minimum value of the similarity. If S<E, we consider PSI=0 because this case corresponds to a very low agreement of the two partitions.

Distance variant of PSI is defined as 1-PSI:

$$PSI_d = \begin{cases} \dfrac{\max(K,K')-S}{\max(K,K')-E} & S \geq E, \max(K,K') > 1 \\ 1 & S < E \\ 0 & K = K' = 1 \end{cases} \tag{19}$$

Value of E depends on the similarity between the structures of two clusterings in terms of number and size of clusters. If the structures are close to each other, $E \approx 1$. Accordingly, simplified variant of PSI is defined by assuming $E=1$:

$$PSI^* = \begin{cases} \dfrac{S-1}{\max(K,K')-1} & S \geq 1, \max(K,K') > 1 \\ 0 & S < 1 \\ 1 & K = K' = 1 \end{cases} \tag{20}$$

6.3 Metric Properties of PSI

The proposed index is normalized in the range of [0, 1] and corrected for chance. In this section, we prove metric property of the distance form of PSI in (19).

Nonnegative: In (19), where $S \geq 1$, $max(K, K') > 1$, since $E \leq 1$, $max(K, K')-E$ is always larger than or equal to 1. The total similarity S equals to $max(K, K')$ only in a perfect match. In all other situations it is less than $max(K, K')$, hence $max(K, K')-S \geq 0$ holds. Therefore, it is true that:

$$PSI_d \geq 0 \tag{21}$$

Symmetric: The similarity of two clusters according to (15) is symmetric. The pairs of clusters are found according to the maximum matching which does not depend on whether we compare P to G or vice versa. Therefore, the total similarity in (14) is symmetric. To derive the expected value of the similarity in (16), we take two largest clusters in P and G as the best match. This action is also independent on the direction of the comparison. According to (18), when the similarity S and its expected value E are symmetric, the whole index is also symmetric:

$$PSI_d(P,G) = PSI_d(G,P) \tag{22}$$

Reflexivity: If $P=G$, the total similarity according to (14) and (15) is $max(K,K')=K$, and therefore $PSI_d=0$. On the other hand, if $PSI_d=0$, it follows that $S=max(K, K')$. This may happen only if the number of clusters is the same and the similarity of every two paired clusters according to (15) is 1. The similarity of two clusters is 1 if and only if they are exactly the same. Therefore, all clusters in P and G must be equal, and accordingly, $P=G$:

$$PSI_d(P,G) = 0 \ \ if\ and\ only\ if \ \ P = G \tag{23}$$

Triangular inequality: In Appendix C, we prove the triangular inequality for the simplified form of PSI in (20). The simplified form is therefore proven to be metric. Experiments for clustering with different structures indicate that the triangular inequality in most cases holds for the original form of PSI as well. However, the term E in the denumerator in (18) makes it difficult to prove in general.

6.4 Other Properties

a) Normalized to the number of clusters

The proposed validity index has low dependency on the number of clusters and this dependency decreases as the number of clusters increases. In (18), the similarity is normalized by $max(K, K')-E$. Because of E, the index is not independent on the number of clusters. However, since $E \leq 1$ and when $max(K, K')$ increases, the impact of E de-

creases.

b) Imbalanced clusters

One important advantage of the proposed index is its independency on the size of clusters because each cluster, either small or large, has equal impact on the similarity value. For example, suppose that in two clusterings, there are two perfect pairs where in one pair the clusters are large and in the other one they are small. Both of them increase the total similarity by the same amount.

7 Experiments

We next evaluate the external validity indexes based on their performance on partitions. To investigate different properties of an index, a variety of partitions should be considered. We provide comparisons with artificially generated partitions to demonstrate whether an index meets the required properties. We also study the effect of dimensionality and cluster overlap.

7.1 Selected Indexes and Artificial Partitions

We compare the proposed index to the state-of-art external indexes. Since all adjusted indexes in the pair-counting group behave similar [23], we use only ARI as the most popular one. Variation of information and mutual information are two representing measures in the information theoretic group. Since NVI_s=AVI_s=NMI=AMI, only NMI is used in the experiments. The performance of arithmetic and geometric mean for normalization of NMI is the same, we therefore employ arithmetic mean only. The normalized Van Dongen criterion, Criterion H and Purity are chosen in the set matching group. The matching in Centroid similarity index depends on the centroids, and therefore, we need real datasets to calculate centroids. However, as we discussed in section 4.2, the results of matching is most likely similar to NVD. We therefore use this assumption in the following, and in these experiments NVD=CSI.

In the test setup, we consider a ground-truth partition *G*, for example with 3000 objects, 1000 objects in each cluster, see Fig. 8, where light grey, grey and black represent the three clusters. In practice, we make an array of the length 3000 with values 1, 2 and 3 representing cluster labels of data. In this case, the first 1000 objects (light grey) have value 1. The partition *P* to be compared with is varied in different ways. The order of the data objects in the two partitions remains the same.

Fig. 8. Two partitions with 3000 objects.

The partitions in the experiment are considered in several aspects: random partitions, the impact of cluster size imbalance, number of clusters and consistency when the error increases in the partitions.

7.2 Random Partitions

Consider partition *P* which consists of random labels as shown in Fig. 9. We conduct experiments for different number of clusters from *K*=1 to 20 in *P*. The indexes NMI, ARI and PSI give values close to zero independent on the number of clusters. The values of the other three indexes are not zero because they are not corrected for chance, see Fig. 10. Normalized mutual information gives zero in this case which shows that NMI has the same performance as adjusted mutual information. This result further verifies our claim in (13).

Fig. 9. Clustering *P* represents a random partition with two clusters.

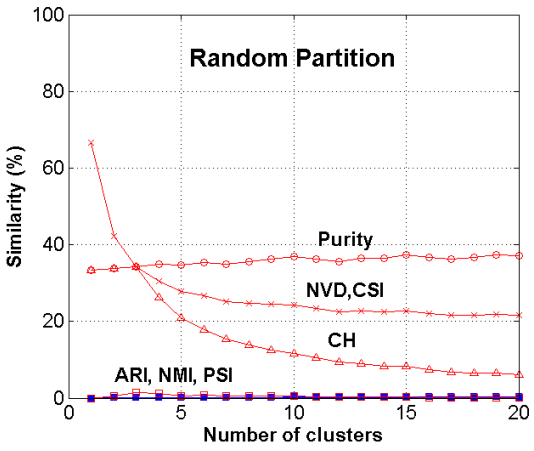


Fig. 10. Random partitioning with different number of clusters in *P* from *K*=1 to 20.

7.3 Monotonicity

We change the partition *P* linearly in three ways and study the response of the indexes.

First we enlarge the first (light grey) cluster in *P* in steps of 50 objects until only one cluster remains, see Fig. 11. Second, we enlarge the grey cluster in the same way, see Fig. 13, and third, we change part of the labels in all clusters of *P* and keep the cluster sizes unchanged, see Fig. 15. In Fig.12, NMI, ARI and NVD have very clear knee points when the light grey cluster reaches 2000 objects because at this point the number of clusters decreases by 1. For NMI and ARI, the index values increase when the cluster size approaches to 2000. In this situation, there are still three clusters and the results indicate that NMI and ARI ignore relatively small clusters and put more weights on large clusters. When the light grey cluster size is 2000, there is a local maximum when the number of clusters changes from three to two. NVD is constant between 1500 to 2000, and 2500 to 3000. The asymmetric matching of clusters in *NVD* causes the problem. Suppose that the size of the grey cluster (*x*) in *P* is less than 500. After matching *P* to *G*, the number of shared objects is 1000+*x*+1000 and *G* to *P* where both light grey and grey clusters in *G* are matched with the light grey one in *P*, the

number of shared objects is 1000+(1000-*x*)+1000. Summing up, the number of shared objects in two directions is 5000 which is independent of *x*. Therefore, when the size of the first cluster is between 1500 and 2000, the similarity remains a constant 5000 / 6000=0.83.

The proposed PSI has near linear dependency on the size of the light grey cluster. The indexes CH and Purity have good linear behavior but including an offset by 33% because they are not corrected for chance. If we made them corrected, the same issues as with the other indexes would appear. Note that Purity does not compare two clusterings in both directions. If we compare *G* to *P* instead of *P* to *G*, the results is different and without linear behavior.

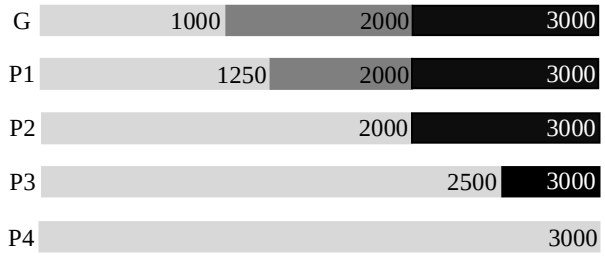


Fig. 11. Enlarging the first (light grey) cluster in steps of 50 objects by moving the objects from the other two clusters.

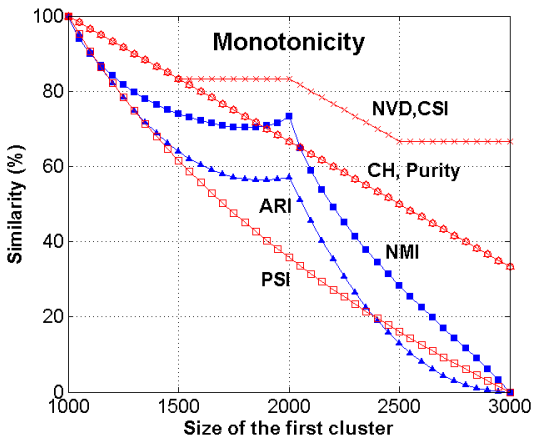


Fig. 12. Increasing the size of the first cluster.

We repeat the experiment by enlarging the size of the second cluster. The difference to the previous case is that the number of clusters remains 3 until the second cluster contains all the objects. The results in Fig. 14 show better performance for NMI and ARI compared to the previous case. The reason is that this time there is no change in the number of clusters in *P*. The same arguments for NVD, CH and CA are valid as for the previous case. The knee point for NVD is where the size of the biggest cluster becomes more than 2000 (compare P_2 and *G* in Fig. 13) and all three clusters of *G* are matched to the grey cluster of *P*. Interesting observation is that PSI results the same curves in both of the cases, which indicates that it depends less on the number and size of clusters than the other indexes.

Next, we change part of the labels in all clusters of *P*. At each step, 50 more objects will be wrongly labeled in each cluster until all objects in *G* are equally distributed among the three clusters in *P*, see Fig. 15.

Fig. 13. Enlarging the second (grey) cluster in steps of 50 objects as in Fig. 11.

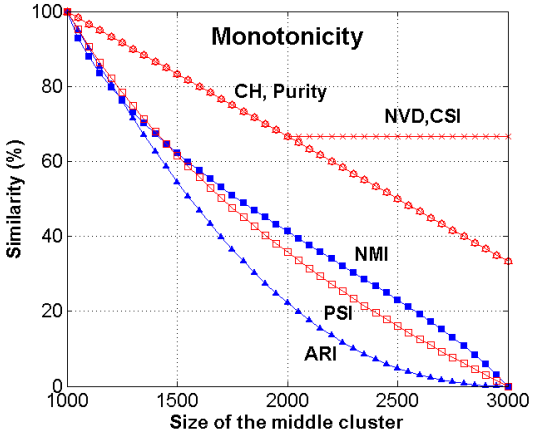


Fig. 14. Increasing the size of the second cluster until it contains all data objects.

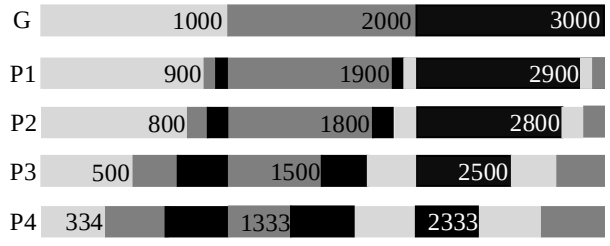


Fig. 15. Increasing the number of incorrectly labeled objects.

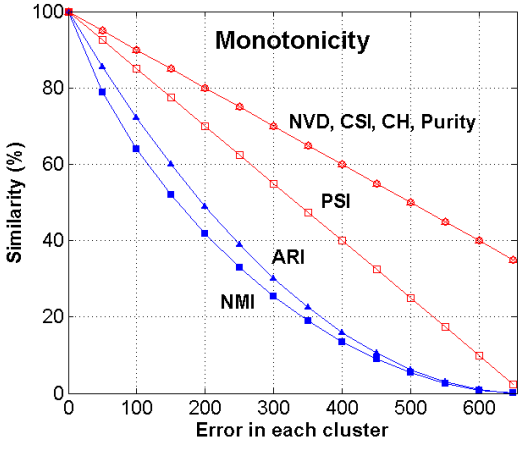


Fig. 16. Increasing the error of each cluster in *P*.

The similarity values of PSI and NVD, CH and Purity decrease linearly but NVD has higher similarity values than PSI, see Fig. 16. Since NVD, CH and Purity are not corrected for chance and are biased to random partitions, they have a higher lower bound. If we made them corrected, they would lose the linearity. The results of NVD, CH and Purity are exactly the same because the matching for all cases in this experiment is the same, which further verifies our claim in (12). Both NMI and ARI have decreasing curves and their values are always lower than those of the set matching indexes. One reason is that NMI and ARI consider also the unmatched parts of clusters.

7.4 Cluster Size Imbalance

In this experiment, we study the impact of cluster size. In Fig. 17, we consider sets of partitions where P_1 and P_2 have 200 objects (20%) wrongly labeled in the first two clusters. The size of the third cluster is decreased from 2000 to 50 in steps of 50.

Since the labels of the first two clusters remain exactly the same, the only difference originates from the size of the third cluster. We assumed that the clusters with different sizes have the same importance, and therefore, the results should be independent of the size of the third cluster. As shown in Fig. 18, all indexes except PSI are affected by the cluster size imbalance. For example, the similarity value of ARI is much lower (66%) when the size becomes 50 than when it is 2000 (91%). The results indicate that most indexes are affected more by the larger clusters. NVD, CSI, CH and Purity values are higher and in a narrower range, which indicates better performance

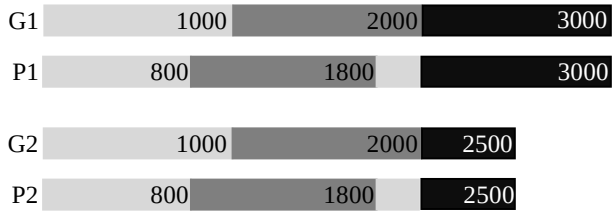


Fig. 17. The effect of cluster size imbalance; same error in the first two clusters and no error in the third clusters.

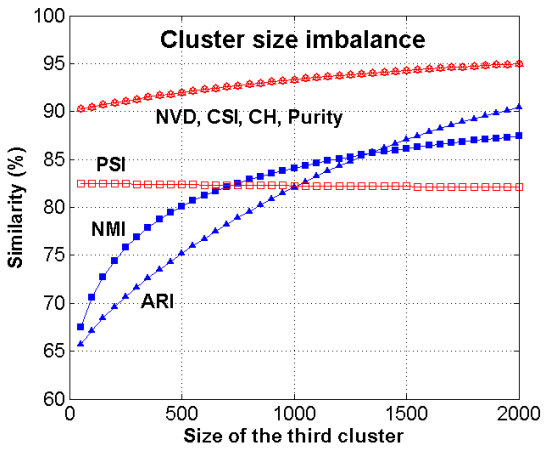


Fig. 18. The effect of cluster size imbalance on the indexes; the partitions contain two clusters with the fixed size and error and the size of the third cluster decreases in steps of 50.

of set matching indexes. Since matching results for NVD, CH and Purity are the same, their results are also the same, see (12). The proposed PSI is the one that copes best with the cluster size imbalance.

7.5 Number of Clusters

We study the effect of the number of clusters by wrongly labeling 200 objects in each cluster and then varying the number of clusters as shown in Fig. 19. The size of clusters is fixed.

The indexes have similar trend on increasing the number of clusters except non-adjusted set matching indexes (NVD, CSI, CH and Purity), see Fig. 20. When increasing the number of clusters, the similarity values rise from as low as 25% up to 80%. However, the impact is much more significant for the small number of clusters from two to four. PSI has better performance than NMI and ARI, but only NVD, CSI, CH and Purity are completely independent on the number of clusters. Considering NVD equation in Table 2 and the same percentage of error across clusters in this experiment, it is trivial to show that NVD is independent on the number of clusters. Since matching results for NVD, CSI, CH and Purity are the same, their results are also the same, see (12). In this experiment, we see that correction for chance has bad effect as it makes the index dependent on the number of clusters. Overall, set matching indexes show better performance than the representatives from pair-counting and information theoretic indexes.

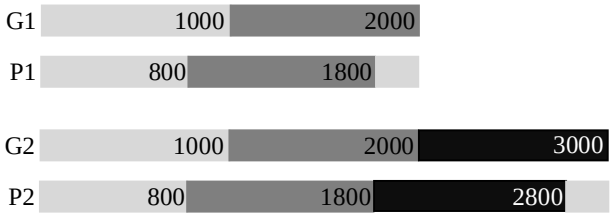


Fig. 19. There are 200 objects wrongly labeled in each cluster and the number of clusters varies.

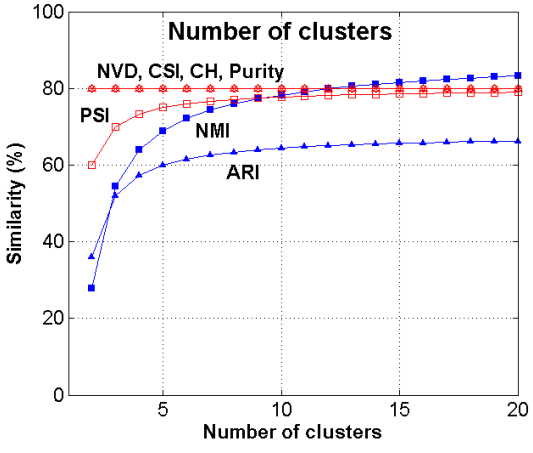


Fig. 20. The effect of number of clusters (K=2 to 20), while the size and error of each cluster are fixed.

7.6 Overlap of clusters

We use a series of data sets (called M2), all containing two clusters (1000 points each) in 8-dimensional space but with varying cluster overlap. The points were generated by Gaussian distribution with the same (constant) variance. The overlap was created by moving one of the clusters closer to the other step by step. The amount of overlap is measured by how many points in a cluster are closer to the centroid of the other cluster than to the centroid of its own cluster.

We cluster these datasets by random swap algorithm [41] and compare the result against ground truth partitions. Fig. 21 shows that all NVD, CSI, CH, Purity, and PSI react as expected. NVD approximately equals to the amount of the overlap, but is lower limited by 0.50. For example, with 15% overlap we expect to have 0.85 similarity. On the other hand, PSI applies correction for chance. Expected similarity of random partition into two clusters is 0.50, and corrected similarity 1-(overlap/0.50), accordingly. With 15% overlap, the expected similarity would be 0.70. The results of PSI are near optimal response (dashed black line).

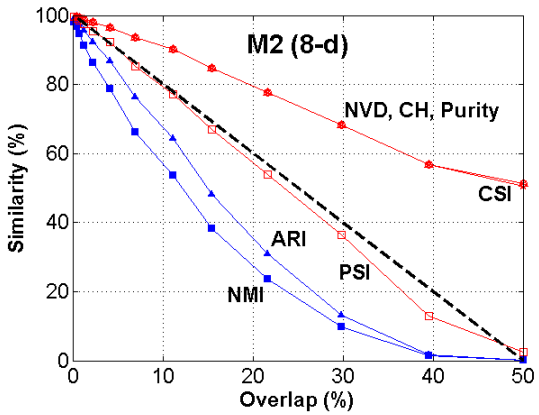


Fig. 21. Effect of the overlap on the similarity measures.

7.7 Dimensionality of data

We used the same M2 data sets but this time we fix the overlap to 15% and vary the dimensionality from 1 to 512. The results in Fig. 22 show that all the methods are invariant to the dimensionality up to a limit (about 256). Decrease of the index values is caused by over-optimization of the clustering algorithm: with high-dimensional data, it can optimize MSE better than would be with the ground truth partition. Otherwise, NVD, CSI, CH, Purity, and PSI again perform as expected with this overlap: NVD gives 0.85 (without) and PSI gives 0.70 (with correction for chance).

7.8 Applications

We study next how the four indexes (ARI, NMI, 1-NVD, PSI) perform with applications. We perform three experiments with the following hypotheses.

In the first experiment, we cluster the dataset Unbalance, see Fig. 23, to k=8 clusters by the following algorithms: random swap (RS) with 5,000 iterations [41], agglomerative clustering with ward criterion (AC), k-means

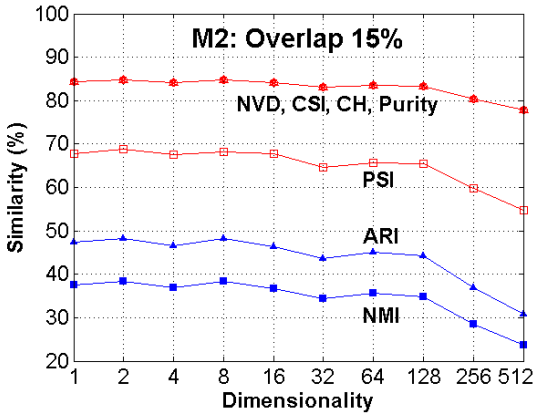


Fig. 22. Effect of dimensionality on the similarity measures.

(KM), and single link (SL). All these methods aim at minimizing total squared error except the single link.

The clusterings are then compared with the known ground truth in Table 6. The result of PSI corresponds best to the expectations: RS and AC are both good at optimizing the structure of the data whereas AC tends to make more point-wise errors at the partition borders. KM detects the dense cluster (2000 points) on the top, but it breaks the two other dense clusters into six smaller subclusters, and merges the five smaller ones (each 100

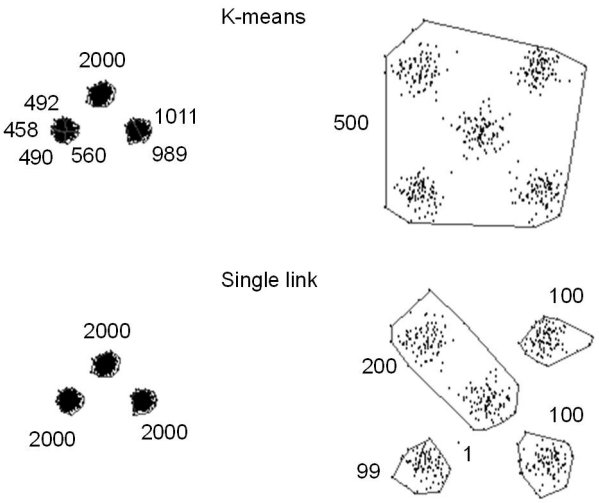


Fig. 23. Clustering results of the data set Unbalance using k-means (above), and single link (below).

TABLE 6
CLUSTERING OF UNBALANCE BY FOUR ALGORITHMS

Algorithms	External indexes			
	ARI	NMI	NVD	PSI
RS	1.00	1.00	1.00	1.00
AC	1.00	1.00	1.00	1.00
SL	1.00	0.99	0.99	0.78
KM	0.66	0.77	0.78	0.18

points) into one cluster, see Fig. 23 (top). All indexes react to these errors but only PSI recognizes that this clustering is off very low quality. SL finds all clusters correctly except that it merges two small ones leaving one orphan point as its tiny cluster, see Fig. 23 (below). Only PSI reacts strongly enough to this situation.

In the second experiment, we take ground truth clusters of the well-known *Yeast* data set (UCI), and then remove the smallest clusters one by one, see Fig. 24. The results in Table 7 show that only PSI provides significant differences due to the cluster removal, mainly because it treats all clusters of equal importance independent of their size.

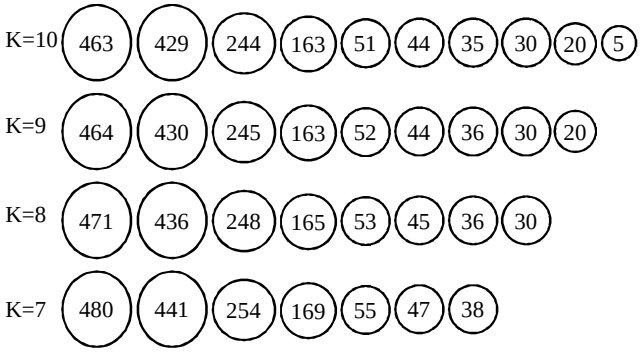


Fig. 24. Removing small clusters one by one and distributing their objects in the other clusters.

TABLE 7
CLUSTERING OF YEAST

Clusters (K)	External indexes			
	ARI	NMI	NVD	PSI
9	1.00	0.99	1.00	0.88
8	0.99	0.97	0.98	0.74
7	0.97	0.93	0.97	0.60

In the third experiment, we study how well the indexes apply for the task of detecting the number of clusters for Unbalance data set that contains eight clusters. We use the stability-based approach in [42] as follows. Ten subsets are generated by random sampling (with the sampling rate 0.2) from the data set. Each subset is then clustered by random swap algorithm with different number of clusters in range $k \in [2, 20]$. The similarity between the clustering of each subset and the clustering of the fullest is calculated by an external index. The stability is then measured as the average index values for all the subsets. The hypothesis is that the correct number of clusters is the one with highest stability (highest index value).

The results in Fig. 25 show that all indexes are applicable to this task, and the bigger problems originate from other factors than the choice of the index. All the indexes show maximum stability with k=8, but the clustering results are also stable with k=2 and k=4. Overall, PSI performs most consistent especially for values $k \geq 5$. In the range of k=5..7, all the indexes except PSI fail to detect high instability in the 5 small-sized clusters.

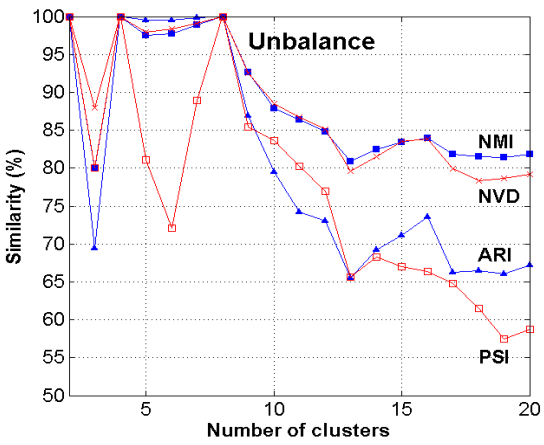


Fig. 25. Solving number of clusters based on stability of clusterings.

8 CONCLUSION

We have conducted a systematic study on existing set matching indexes by analyzing them in three different aspects: similarity measure of two clusters, matching the clusters, and the overall summation. We have shown that the difference between NVD, CH, Purity and CSI is only about their matching. If their matching result were the same, all these indexes would provide equivalent result. We have also pointed out that Purity and the measures cited as classification error or classification accuracy are equivalent.

We defined concrete requirements that an external index should meet, and introduced new arrangement of experiments based on synthetic data that can be used for systematic evaluation of any index according to these criteria. According to our experiments, set matching indexes perform better than the selected indexes of pair-counting and information theoretic indexes in many aspects such as cluster size imbalance, number of clusters and linear changes.

None of the existing set matching measures use correction for chance, and they also normalize the index across all data points. Based on these observations, we propose a new index called PSI that applies correction for chance, and performs normalization for each cluster separately. We show that the simplified form of PSI is a metric.

For the information theoretic measures, we have also shown that NMI=AMI=NVIs=AVIs under hypergeometric distribution assumption, which was also verified by our experiments.

ACKNOWLEDGMENT

This research has been supported by MOPIS project and partially by Nokia Foundation grant.

REFERENCES

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

[2] J. Handl, J. Knowles and D.B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, 21(15), pp. 3201-3212, 2005.

[3] G.W. Milligan and M.C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, 50(2), pp. 159–179, 1985.
[4] E. Dimitriadou, S. Dolnicar and A. Weingassel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika*, 67(1), pp. 137–160, 2002.
[5] M. Halkidi, Y. Batistakis and M. Vazirgiannis, "On clustering validation techniques," *J. Intelligent Information Systems*, 17(2-3), pp. 107–145, 2001.
[6] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(12), pp. 1650–1654, 2002.
[7] J. Wu, H. Xiong and J. Chen, "Adapting the right measures for k-means clustering," *15th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'09)*, pp. 877–886, 2009.
[8] J. Wu, J. Chen, H. Xiong and M. Xie, "External validation measures for k-means clustering: A data distribution perspective," *Expert Systems with Applications*, 36(3), pp. 6050–6061, 2009.
[9] N.X. Vinh, J. Epps and J. Bailey, "Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance," *J. Machine Learning Research*, 11, pp. 2837–2854, 2010.
[10] A. Strehl, J. Ghosh and C. Cardie, "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," *J. Machine Learning Research*, 3, pp. 583-617, 2003.
[11] L.I. Kuncheva and D.P. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11), pp. 1798–1808, 2006.
[12] S. Zhang, H. Wong and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognition*, 45(6), pp. 2214–2226, 2012.
[13] L. I. Kuncheva, S. T. Hadjitodorov and L. P. Todorova, "Experimental comparison of cluster ensemble methods," *9th Int. Conf. on Information Fusion*, pp. 1-7, 2006.
[14] P. Fränti, J. Kivijärvi, T. Kaukoranta and O. Nevalainen, "Genetic algorithms for large scale clustering problems," *The Computer Journal*, 40 (9), pp. 547-554, 1997.
[15] I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J.M. Pérez and J. Martín, "Towards a standard methodology to evaluate internal cluster validity indices," *Pattern Recognition Letters*, 32, pp. 505-515, 2011.
[16] M.H.C. Law and A.K. Jain, "Cluster validity by bootstrapping partitions," *Technical Report MSU-CSE-03-5*, Dept. of Computer Science and Engineering, MSU, Michigan, USA, 2003.
[17] M. Falasconi, A. Gutierrez, M. Pardo, G. Sberveglieri and S. Marco, "A stability based validity method for fuzzy clustering," *Pattern Recognition*, 43(4), pp. 1292-1305, 2010.
[18] D. Pascual, F. Pla, and J.S. Sanchez, "Cluster validation using information stability measures," *Pattern Recognition Letters*, 31(6), pp. 454-461, 2010.
[19] B.E. Dom, "An information-theoretic external cluster-validity measure," *Research Report RJ 10219*, IBM, 2001.
[20] P. Fränti, M. Rezaei and Q. Zhao, "Centroid index: cluster level similarity measure," *Pattern Recognition,* 47(9), pp. 3034-3045, 2014.
[21] N.X. Vinh, J. Epps and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?," *26th Int. Conf. on Machine Learning (ICML'09)*, pp. 1073-1080, 2009.
[22] S. Wagner, D. Wagner, "Comparing clusterings – an overview," *Technical Report, 2006-4*, Fakultät für Informatik, Universit¨at Karlsruhe (TH) , 2006.
[23] A.N. Albatineh, M. Niewiadomska-Bugaj and D. Mihalko, "On similarity indices and correction for chance agreement," *J. Classification*, 23(2), pp. 301-313, 2006.
[24] M. Meila, "Comparing clusterings – an information based distance," *J. Multivariate Analysis*, 98(5), pp. 873-895, 2007.
[25] T.O. Kvalseth, "Entropy and correlation: some comments," *IEEE Trans. Syst. Man Cybern.*, 17(3), pp. 517–519, 1987.
[26] M.C.P. de Souto, A.L.V. Coelho, K. Faceli, T.C. Sakata, V. Bonadia and I.G. Costa, "A comparison of external clustering evaluation indices in the context of imbalanced data sets," *2012 Brazilian Symposium on Neural Networks*, pp. 49-54, 2012.
[27] M. Meila and D. Heckerman, "An experimental comparison of model based clustering methods," *Machine Learning*, 41(1-2), pp. 9–29, 2001.
[28] S.V. Dongen, "Performance criteria for graph clustering and Markov cluster experiments," *Technical Report INSR0012*, Centrum voor Wiskunde en Informatica, 2000.
[29] Q. Zhao and P. Fränti, "Centroid ratio for a pairwise random swap clustering algorithm," *IEEE Trans. Knowledge and Data Engineering*, 26(5), pp. 1090-1101, 2014.
[30] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, pp. 193–218, 1985.
[31] A.A. Goshtasby, "Similarity and dissimilarity measures," in *Image Registration - Principles, Tools and Methods. Advances in Computer Vision and Pattern Recognition*, pp. 7-66, Springer London, 2012.
[32] S. Theodoridis, K. Koutroumbas, "Clustering: basic concepts," Pattern Recognition 4th edn, Academic Press, New York, pp. 595, 624, 2009.
[33] E. Rendon, I. Abundez, A. Arizmendi, E.M. Quiroz, "Internal versus external cluster validation indexes," *International Journal of Computers and Communications* 5(1), pp. 27-34, 2011.
[34] N. Nguyen and R. Caruana, "Consensus Clusterings," *IEEE Int. Conf. Data Mining*, pp. 607-612, 2007.
[35] N. Iam-On, T. Boongoen, S. Garrett, C. Price, "A Link-Based Approach to the Cluster Ensemble Problem," *IEEE Trans. Pattern Analysis and Machine Intelligence* 33(12) pp. 2396-2409, 2011.
[36] S. Choi, S. Cha and C. Tappert, "A survey of binary similarity and distance measures," *J. Systemics, Cybernetics and Informatics* 8(1), pp. 43-48, 2010.
[37] SB. Dalirsefat, A. Meyer and SZ. Mirhoseini, "Comparison of similarity coefficients used for cluster analysis with amplified fragment length polymorphism markers in the silkworm, Bombyx mori," *J. Insect Science*, 9, pp. 681-689, 2009.
[38] B. Sarker, "The resemblance coefficients in group technology: A survey and comparative study of relational metrics," *Computers and Industrial Engineering*, 30, pp. 103–116, 1996.
[39] B. Larsen, C. Aone, "Fast and effective text mining using linear time document clustering," *5th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, pp. 16–22, 1999.
[40] H. W. Kuhn, "The Hungarian method for the assignment problem," *50 Years of Integer Programming 1958–2008*, pp. 29–47, 2010.
[41] P. Fränti and J. Kivijärvi, "Randomised local search algorithm for the clustering problem," *Pattern Analysis & Applications*, *3*(4), pp. 358-369, 2000.
[42] E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity," *Neural computation* 13(11), pp. 2573-2593, 2001.

Mohammad Rezaei received his BSc degree in Electronic engineering in 1996 and his M.Sc. degree in biomedical engineering in 2003 both from Amirkabir university of Technology, Tehran, Iran. Currently he is a PHD student in the university of Eastern Finland. His research interests include data clustering, multimedia processing, classification and retrieval.

Pasi Fränti received his MSc and PhD degrees from the University of Turku, 1991 and 1994 in Science. Since 2000, he has been a professor of Computer Science at the University of Eastern Finland. He has published 65 journals and 145 peer review conference papers, including 13 IEEE transaction papers. His research interests include clustering algorithms and location-based systems.

APPENDIX A

Proof of Theorem 1

First, we introduce a new way to derive the expected value of mutual information in case of random partitions and under hyper-geometric distribution assumption and then we use the expected value to prove (13). Consider a pair of clusters P_i and G_j. The probability that an object in P_i exists in G_j is m_j / N. Accordingly, the number of objects in both P_i and G_j is simplified as: $n_{ij}=n_i \times (m_j / N)$. Then, the expected value can be calculated according to (7) as:

$$E(MI) = E\left\{\sum_i \sum_j \frac{n_{ij}}{N} \log\left(\frac{N \times (n_i \times m_j / N)}{n_i \times m_j}\right)\right\}$$
$$= E\left\{\sum_i \sum_j \frac{n_{ij}}{N} \log(1)\right\} = 0 \quad (24)$$

According to (2), AMI=NMI which confirms the result from [9]. Applying $max(MI)=(H(P) + H(G))/2$ as an option for normalization [22], [17], we can write:

$$AMI = NMI = \frac{2 \times MI(P,G)}{H(P)+H(G)} \quad (25)$$

Since $E(H(P)) = H(P)$ and $E(H(G)) = H(G)$ under hyper-geometric distribution assumption, the expected value of VI (8) is derived as:

$$E(VI) = H(P) + H(G) \quad (26)$$

VI is a dissimilarity measure and $min(VI) = 0$ when the two partitions are equal. Therefore, the adjusted variation of information according to (2) is:

$$AVI = \frac{VI}{H(P)+H(G)} \quad (27)$$

An upper bound for VI is $H(P) + H(G)$ and therefore (27) also represents the normalized variation of information. We simplify AVI_s and NVI_s using (8) as follows:

$$AVI_s = NVI_s = \frac{2 \times MI(P,G)}{H(P)+H(G)} \quad (28)$$

From (25) and (28), we see that the adjusted mutual information and adjusted variation of information are equal to their normalized forms, and thus, theorem 1 is proven.

APPENDIX B

Proof of Theorem 2

Suppose that in a matching, m_1 is paired to $n_i < n_1$ and n_1 is paired to $m_j < m_1$ (case *a*). We show that if we change the matching so that m_1 is paired to n_1 and m_j is paired to n_i (case *b*), higher similarity is achieved. The total similarities for these two cases (*a* and *b*) are:

$$S_a = \frac{m_1 \times (n_i / N))}{\max(m_1, n_i)} + \frac{m_j \times (n_1 / N))}{\max(m_j, n_1)}$$
$$S_b = \frac{m_1 \times (n_1 / N))}{\max(m_1, n_1)} + \frac{m_j \times (n_i / N))}{\max(m_j, n_i)} \quad (29)$$

where S_a is the original pairing and S_b is the new pairing after changing the pairs for m_1 and m_j. Six different situations may happen:

1. $m_1 > m_j > n_1 > n_i$

$$\left[S_a = \frac{1}{N}(n_1 + n_i)\right] = \left[S_b = \frac{1}{N}(n_1 + n_i)\right]$$

2. $m_1 > n_1 > m_j > n_i$

$$\left[S_a = \frac{1}{N}(n_i + m_j)\right] < \left[S_b = \frac{1}{N}(n_1 + n_i)\right]$$

3. $m_1 > n_1 > n_i > m_j$

$$\left[S_a = \frac{1}{N}(n_i + m_j)\right] < \left[S_b = \frac{1}{N}(n_1 + m_j)\right]$$

4. $n_1 > m_1 > n_i > m_j$

$$\left[S_a = \frac{1}{N}(n_i + m_j)\right] < \left[S_b = \frac{1}{N}(m_1 + m_j)\right]$$

5. $n_1 > n_i > m_1 > m_j$

$$\left[S_a = \frac{1}{N}(m_1 + m_j)\right] = \left[S_b = \frac{1}{N}(m_1 + m_j)\right]$$

6. $n_1 > m_1 > m_j > n_i$

$$\left[S_a = \frac{1}{N}(m_j + n_i)\right] < \left[S_b = \frac{1}{N}(m_1 + n_i)\right] \quad (30)$$

Considering all the above situations, pairings (m_1, n_i) and (n_1, m_j) must be changed to (n_1, m_1) and (m_j, n_i) to achieve higher similarity. We can apply this proof recursively to all the smaller clusters as well. Hence, the two largest clusters must be always paired and then the next two largest and so on in order to achieve maximum total similarity with a random partition. This proves the theorem 2.

APPENDIX C

Triangular Inequality Proof for the Simplified form of PSI

Let P_1, P_2 and P_3 be three partitions with K_1, K_2 and K_3 clusters, and $K_{12}=max(K_1,K_2)$, $K_{23}=max(K_2,K_3)$, $K_{13}=max(K_1,K_3)$. Let n_i, n_j and n_k be the number of objects in clusters *i*, *j* and *k* in P_1, P_2 and P_3 respectively. We denote the number of shared objects between clusters by n_{ij}, n_{jk} and n_{ik}. The simplified distance form of PSI, for P_1 and P_2, according to (20) is:

$$D_{12} = \frac{K_{12} - S_{12}}{K_{12} - 1}$$

Lemma. $D_{12}+D_{23} \geq D_{13}$ (31)

Proof. We define $D'_{12} = K_{12} - S_{12}$, $D'_{23} = K_{23} - S_{23}$ and $D'_{13} = K_{13} - S_{13}$ and prove first that: $D'_{12}+D'_{23} \geq D'_{13}$ which is equivalent to

$$K_{12} - S_{12} + K_{23} - S_{23} \geq K_{13} - S_{13} \quad (32)$$

We consider three possible situations and simplify (32):

(1) $K_1 \geq K_{23}$: $S_{12} + S_{23} \leq K_{23} + S_{13}$
(2) $K_3 \geq K_{12}$: $S_{12} + S_{23} \leq K_{12} + S_{13}$
(3) $K_2 \geq K_{13}$: $S_{12} + S_{23} \leq K_2 + (K_2 - K_{13}) + S_{13}$

In the case (3), since $K_2 \geq K_{13}$, it is sufficient to prove $S_{12} + S_{23} \leq K_2 + S_{13}$. Since $K_{23} \geq K_2$ and $K_{12} \geq K_2$, for all cases it is sufficient to prove:

$$S_{12} + S_{23} \le K_2 + S_{13} \tag{33}$$

According to the definitions (14) and (15), we divide the inequality (33) into K_2 sub-inequalities by considering each cluster j in P_2 on the left. Each sub-inequality is of the form:

$$\frac{n_{ij}}{\max(n_i, n_j)} + \frac{n_{jk}}{\max(n_j, n_k)} \le 1 + \frac{n_{ik}}{\max(n_i, n_k)} \tag{34}$$

Clusters *i* and *k* from P_1 and P_3 which are the pairs for cluster *j* are not necessarily a pair in comparing P_1 and P_3. Since S_{13} is derived according to perfect matching, we can consider another matching of P_1 and P_3 in which *i* and *k* are paired. If (33) holds in this case, it will also be true for S_{13} which is the maximum possible similarity.

If the cluster *j* has a pair cluster only in P_1 or P_3, it is trivial to prove (34). If it has pair clusters both in P_1 and P_3, and $n_{ij} + n_{jk} \le n_j$, proving (34) is trivial as well since the left side of the inequality is smaller than one. Note that if the clusters *i* and *k* do not have any shared objects, $n_{ij} + n_{jk} \le n_j$. So we prove (34) when $n_{ij} + n_{jk} > n_j$. Considering a minimum value for n_{ik} as $n_{ij} + n_{jk} - n_j$, we rewrite (34) as follows:

$$\frac{n_{ij}}{\max(n_i, n_j)} + \frac{n_{jk}}{\max(n_j, n_k)} \le 1 + \frac{n_{ij} + n_{jk} - n_j}{\max(n_i, n_k)} \tag{35}$$

Three possible cases are:

(1) $n_j \ge max(n_i, n_k)$: By replacing $max(n_i, n_j)$ and $max(n_j, n_k)$ by n_j and after simplifications, we have:

$(n_{ij}+n_{jk}-n_j)(n_j-\max(n_i,n_k)) \ge 0$

which is always true in this case.

(2) $n_i \ge max(n_j, n_k)$: We replace $max(n_i, n_j)$ and $max(n_i, n_k)$ by n_i. Since $max(n_j, n_k) \ge n_j$, it is sufficient to prove (35) by replacing $max(n_j, n_k)$ by n_j. The equivalent inequality derived after simplification:

$(n_i - n_j)(n_j - n_{jk}) \ge 0$

is always true.

(3) $n_k \ge max(n_i, n_j)$: The same proof in the case (2) can be applied.

The lemma (31) can now be represented as:

$$\frac{K_{12} - S_{12}}{K_{12} - 1} + \frac{K_{23} - S_{23}}{K_{23} - 1} \ge \frac{K_{13} - S_{13}}{K_{13} - 1} \tag{36}$$

We consider three possible cases:

(1) $K_1 \ge K_{23}$: It is sufficient to prove (36) if K_{23} in denumerator is replaced by K_1. So we simplify (36) as follows:

$$\frac{K_{12} - S_{12}}{K_1 - 1} + \frac{K_{23} - S_{23}}{K_1 - 1} \ge \frac{K_{13} - S_{13}}{K_1 - 1}$$

Since $K_1 \ge 2$, The denumerators can be canceled and the inequality is true according to (32).

(2) $K_3 \ge K_{12}$: The same inference as the case (1) can be performed by replacing K_{12} with K_3.

(3) $K_2 \ge K_{13}$: By simplifying (36), the following equivalent inequality is resulted:

$$S_{12} + S_{23} \le 2K_2 - \frac{(K_{13} - S_{13})(K_2 - 1)}{K_{13} - 1} \tag{37}$$

Using (32), it is sufficient to prove:

$$K_2 + S_{13} \le 2K_2 - \frac{(K_{13} - S_{13})(K_2 - 1)}{K_{13} - 1}$$

After simplification we have:

$$S_{13}(K_2 - K_{13}) \ge (K_2 - K_{13})$$

According to (14), $S_{13} \ge 0$, and therefore the above inequality is true.

According to the cases (1), (2) and (3), the inequalities (36) and consequently (31) hold, thus, the lemma is proven.

Paper P3

M. Rezaei and P. Fränti,
“Can number of clusters be solved by external validity index”,
(Submitted), 2016.

Can the Number of Clusters Be Solved by External Index?

Mohammad Rezaei and Pasi Fränti

Speech & Image Processing Unit
Department of Computer Science, University of Eastern Finland
Joensuu, FINLAND
rezaei@cs.joensuu.fi

Abstract: External indices have been used in the literature for solving the number of clusters but with contradicting results. The idea is to measure stability of the clustering by external validity index when adding randomness to the process. The hypothesis is that the clustering is more stable with the correct number of clusters. In this paper, we study the main components of the stability-based approach. We will discuss how to create the randomness to the process, how to perform the cross-validation, which clustering algorithm, and which external index to apply. We will show that the number of clusters can be solved reliably only when the type of data is known, and all components of the approach are chosen accordingly. We conclude that the stability-based approach is not reliable in practice, and therefore, do not recommend it to be used.

Keywords: Clustering, cluster validation, stability, number of clusters, external index, resampling.

1. Introduction

The number of clusters can be solved by comparing cluster validity of different number of clusters using *internal indices*. They are usually based on two measures: compactness and separation. Compactness measures how similar the objects within the same cluster are, and separation measures how dissimilar the clusters are. For example, several sum-of-square error indices calculate the ratio of within cluster variance and between cluster variance [1]. The main characteristic of the indices is that they use no priori information of the data. A number of indices have been compared in [2], [3], [4], [5] but none has reached a clear state-of-the-art status that would work for a wide range of datasets.

External indices compare the clustering solution to a ground truth data [6], [7], [8], [2], which can be used to study the performance of clustering methods with artificial data. External indices are also suitable to compare two clustering solutions of the same dataset to evaluate difference of the algorithms [9], [10], or utilized in ensemble clustering [11], [10], [12]. Some authors consider two types of the indices: relative index for comparing two clusterings, and external index for comparing a clustering with the ground truth. We consider here both of them as external index.

External indices have also been used for solving the number of clusters [9], [13], [14], [6], [15], [16], [17]. The idea is to generate randomness in the process by resampling the data, cluster the subsamples with a varying number of clusters, and then measure the stability with the presence of the randomness [14]. The stability is measured by calculating the similarity of the clustered subsamples using an external index. The hypothesis is that the clustering is more stable (higher similarity) when having the correct number of clusters. This approach includes the following design choices:

1. Adding randomness
2. Cross-validation strategy
3. Selection of the external index
4. Selection of the clustering method

Randomness is typically created through sub-sampling. The size and number of subsamples are parameters but they are quite straightforward to set except when the size of data is very small. Alternative approaches are to use a randomized algorithm such as k-means with random initialization [18], or by adding noise to the data [19], [20]. However, we will show that k-means itself is unstable and therefore not reliable for this purpose. Adding noise, on the other hand, would require additional noise parameters that are not trivial to set. It might therefore create unexpected artifacts if not properly designed.

Most external indices are restricted to compare partitions of exactly the same data. A straightforward approach [13], [14], [21] is to compare the clustering of a subset to that of its full set but restricting only to the points that are in the subset. Another approach predicts the partition labels of the rest of the points by nearest neighbor mapping using cluster centroids, or by applying a more complicated classifier process [9], [22], [23], [24]. We will also consider comparing the subsets directly by using centroid index [25], which does not require the partition of the data.

The third design choice is to select an external validity index. Rand index [26] gives poor results according to our tests so we consider several other alternatives. Adjusted Rand index [27], information theoretic measures [28], [29] and selected set-matching methods [30], [25], [31] are all suitable for the task. We will show by experiments that the exact choice of the measure is less important, but how it is applied matters much more. All existing methods simply select the number of clusters that provide maximum stability (global maximum), but we will show by counter-examples and experiments that it is better to choose the last local maximum.

The last design choice is the selection of the clustering algorithm. K-means is commonly used but it is highly unstable and not useful in this task. Another more robust algorithm, such as agglomerative clustering [32], random swap [33] or genetic algorithm [34] should be chosen. Another question is which cost function (cluster model) the algorithm should optimize. If we apply squared error criterion but the data is not spherical, we can get clustering that does not fit the data. In principle, we should still find the number of clusters that best fits to this model. However, the stability assumption does not always hold in this mismatch case.

In this paper, we review the literature and provide a systematic study on stability-based methods for solving the number of clusters by external indices. We first introduce the stability-based approach. We then study the design choices for every component and show their limitations. We study how the choice of the clustering algorithm affects the result, and compare the performance of several external indices. We also compare the cross-validation and classification-based approaches. We will show by counter-examples that the maximum stability is not always at the correct number of clusters. We propose a more robust criterion called last local maximum.

To sum up, we will answer whether external stability is applicable for solving the number of clusters, and how it should be done exactly.

2. Stability-Based Method

Given a dataset X with N objects in d dimensions, an external index I, and a clustering algorithm A, the goal is to find the number of clusters K that best fits the structure in the dataset. Clustering is defined *stable* if it remains the same when applied for several datasets generated with the same process or from the same underlying model [18]. The

similarity between every two clusterings is measured by an external validity index. It is expected that the most stable result would be achieved when the correct number of clusters is applied [22].

The idea is demonstrated in Figure 1. Centroid-based clustering is applied to the dataset with five clusters, and to its subset with parameters k=5 and k=8. The clustering results of the full set and the subset are similar when k=5, whereas there are disagreements when k=8. In the full set, the top leftmost cluster is divided whereas in the subset it remains as one; and vice versa, in the top rightmost cluster it is divided in the subset clustering.

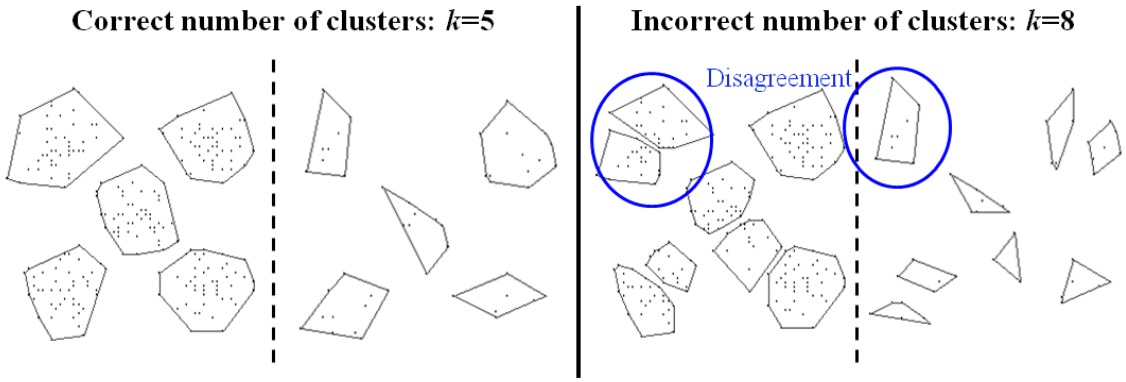


Figure 1. Stability-based method for finding number of clusters; Stable (left) and unstable (right) results are produced if correct and incorrect number of clusters are applied.

Stability, however, can also be achieved with fewer clusters when their positioning is not symmetric [35]. Figure 2 shows two datasets with three well-separated clusters, first with a symmetric (left) and second with a non-symmetric (right) positioning of the clusters. Applying clustering with k=2 gives unstable result for the first dataset but stable results for the second dataset. The second dataset is also stable for k=3, which is the correct number of clusters. Therefore, it is better to select the highest number of clusters that leads to a stable result.

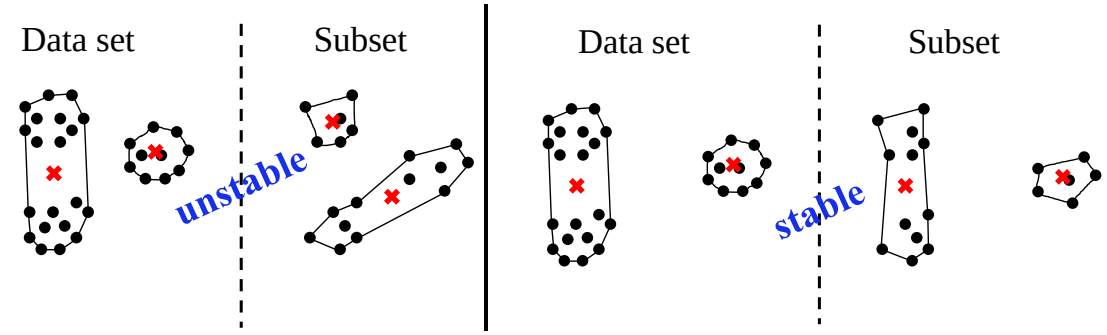


Figure 2. Unstable results for symmetrically positioned clusters (left), and stable results for non-symmetrically positioned clusters (right) when the incorrect number of clusters k=2 is applied.

2.1. Adding randomness

Randomness can be created by one of the following ways:

1. Random sub-sampling [14], [16], [24]
2. Adding random noise [19], [20]

3. Randomizing the algorithm [18]

Most common approach is to create a number of subsets by Monte Carlo sub-sampling [36], where the size and number of the subsets are parameters. The size of subsets should be high but not too close to 100% in order to create significant variation between the subsets. Otherwise, the clustering algorithm may produce always the same result and being always stable [18]. Too low sampling rate, on the other hand, can break the structure in the data as shown in Figure 3. Unless otherwise noted, we use 20% throughout this paper.

The second approach is to add noise to the data by perturbing each individual data object [19], [20]. A noise vector with random orientation is generated but its magnitude depends on the data and is not trivial to set. In [19], the magnitude of noise is derived based on k-nearest neighbors. In [20], a random Gaussian noise with zero mean and fixed standard deviation 0.15 is added to the data. The standard deviation was estimated according to the median standard deviation of the log-ratios for single genes. In the case of categorical data, noising can become complicated. Changing just one attribute randomly might result in an impossible combination of the attributes. We do not consider this approach further.

The third approach is to randomize the algorithm. Randomness of k-means initialization was studied extensively in [18]. It was observed that the clustering result tend to be unstable when too many clusters, and stable with high probability when the correct number of clusters is applied. In the case of too few clusters, both stable and unstable situations were reported, similarly as in Figure 2. However, these analyses were conducted using a better algorithm than the standard random initialization of k-means. This was reasoned that an inconsistent clustering algorithm is completely unreliable and should not be used. We fully agree with this and our observations support it; k-means is not suitable for the randomization but another more stable algorithm could be used.

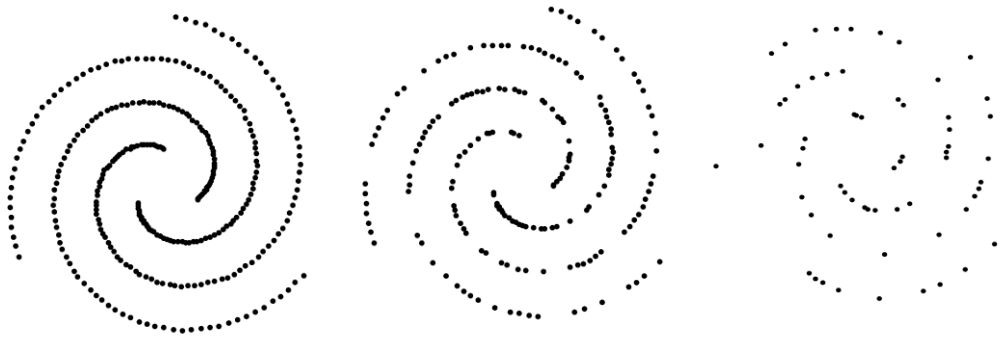


Figure 3. Dataset *spirals* and its subsets with 60% (middle) and 20% (right) subsampling

2.2. Cross-validation strategy

Depending on the randomization strategy, there are several alternatives how to compare the clustering results. If we use the noising or randomized algorithm approaches, we can compare the full sets directly using any external index value. If the sub-sampling approach is selected, there are some limitations on what to compare.

Sub-sampling produces subsets with different sets of points. Most external indices are based on point-level operations and cannot therefore be applied directly because they require having exactly the same set of points. It is possible to limit the comparison to

the points that are in the both subsets. However, the danger is that a too small size of the intersection may not reflect the real similarity of the subsets. With an 80% sub-sampling rate, we have 0.8 × 0.8 = 64% shared points, but with a 20% rate only 4%.

The second solution is to apply a cluster-level index such as *centroid index* [25], which is independent of the data used to produce the clustering. It analyzes how many centroids are differently allocated in the two solutions. It produces clear CI=0 value when the clustering structures are identical. It is directly applicable with any model-based clustering, and is applied to all of the randomization strategies discussed.

The third solution is to predict the missing partition labels in the full set by nearest neighbor mapping based on the cluster centroids [15]. After that, the clustering of any subset can be compared against any another. Even simpler variant is to compare the clustering result of a subset to that of the full set by limiting the comparison to the points that are in the subset [21], [13], [14]. This approach is the most popular in the literature, and we use it as our baseline in the rest of the paper. We refer to these approaches as *cross-validation strategy*. The baseline variant using the sub-sampling strategy is outlined in Figure 4.

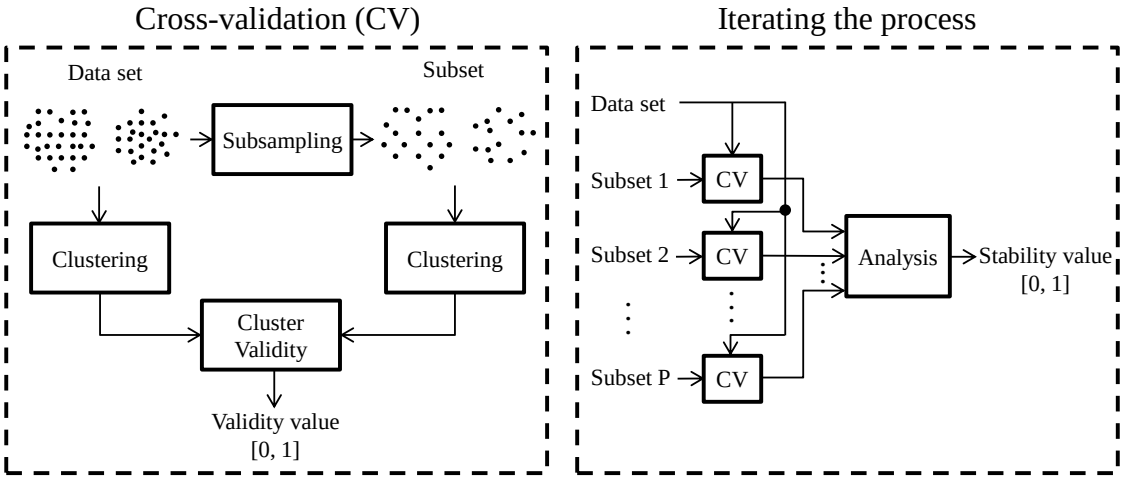


Figure 4. Cross-validation technique; clustering of a full dataset is compared with the clustering of its subset (left). The process is repeated for a number of subsets (right).

2.3. Selecting the number of clusters

Most external indices return a similarity value between the range [0, 1]. We study next how we can conclude from these values that two clustering results are the same, and that the clustering for the given value k is therefore stable. In the following, we consider three approaches (two global, one local) how to select the best k:

1. Maximum stability (global)
2. Normalized maximum stability (global)
3. Last local maximum

The cross-validation approach is repeated by applying clustering with all potential number of clusters $k \in [k_{\min}, k_{\max}]$. We denote the mean value of the validity index for k clusters as I_k. Maximum stability approach uses this mean value as such to indicate the correct number of clusters:

$$K = \arg\max_{k}(I_k) \quad (1)$$

Normalized maximum stability approach (see Figure 5) selects the number of clusters as the maximum difference in mean stability values of the data (I) and the corresponding value (I_0) of the null reference, which is a random dataset drawn from the original data (this is discussed in more detail in section 2.4) [9], [13]:

$$K = \arg\max_k (I_k - I_k^0) \tag{2}$$

This approach is referred to as normalization with regard to the number of clusters [18]. The reason is that the stability value depends on k regardless of the underlying data structure. For example, the stability of clustering for a random uniform dataset decreases as the number of clusters increases. This bias should be removed, and then the same equation (1) should be used. The same approach is also used in the gap statistics [37].

In [18], it was observed that clustering can also be stable when having too few clusters but rarely when too many clusters. It is therefore possible to find several maxima of the index, and taking the global maximum might provide wrong result. We therefore consider *last local maximum* as a new criterion in this paper. For this, a threshold (I_{th}) is set to decide how high an index value is considered stable. The selection now becomes:

$$K = \arg\max_k (k | I_k > I_{th}) \tag{3}$$

In the case of centroid index [25], we can interpret CI=0 for stable and CI>0 for unstable clustering. For all other indices, we set threshold value 0.9 throughout this paper. This selection seems robust for the datasets used in this paper, but the downside is that it adds one more control parameter to the process.

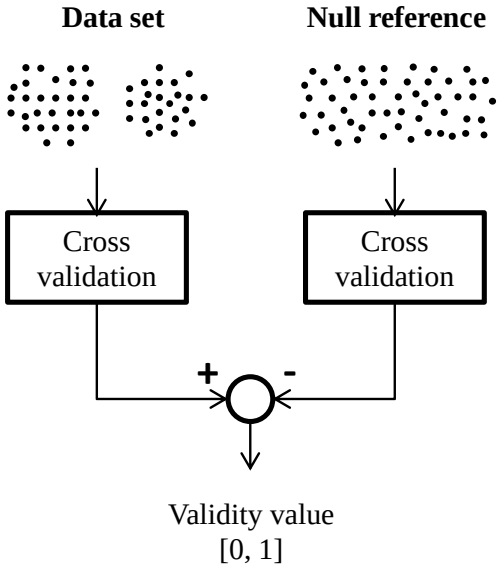


Figure 5. Finding the strongest evidence against null hypothesis by evaluating the difference between mean index values for a dataset and its null reference

2.4. Null hypothesis

We study next the theoretical background of the normalization to better understand why it has been considered. Originally *Null hypothesis* H_0 assumes that the data is random and there are no clusters: K=1 [8], [9], [36]. Acceptance or rejection of this hypothesis is based on statistical inference. The alternative hypothesis H_1 assumes a specific structure in the dataset, for example, K=3.

In the stability-based method, H_1 corresponds to X, and H_0 corresponds to a *null reference* X_0, which is a uniform random dataset having the same parameters and dimension as X [9], see Figure 6. The null reference is generated by randomly sampling points in the range of the attributes of the original dataset. Sometimes, only the relationships between data objects are available by a similarity matrix. In this case, the null reference is produced by randomly generating the matrix with the values in the range of minimum and maximum similarity values between the objects in the original dataset [38].

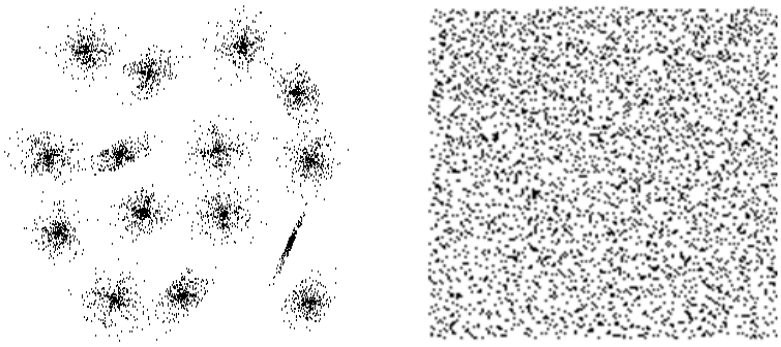


Figure 6. Dataset S1 (left), the corresponding null references (right).

Cross-validation is performed both for X and its null reference X_0 separately using a large number of repeats. This results in two probability density functions (PDF) of the index I, corresponding to H_0 and H_1. These are considered as random variables, see Figure 7 for a theoretical example. The goal is to analyze whether there is statistically significant evidence that the two distributions are different.

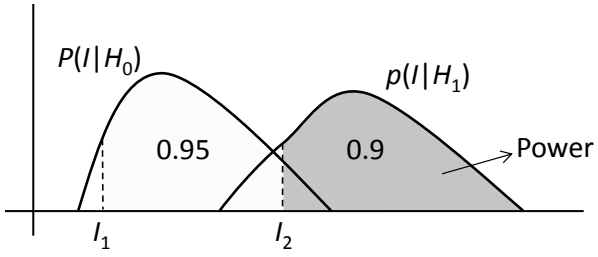


Figure 7. Hypothesis testing; PDF of H_0 corresponds to X_0 and PDF of H_1 corresponds to $\boldsymbol{X}$.

Practical example is shown in Figure 8. We generated a uniform null reference for the dataset in Figure 1, and produced 100 subsets with the sampling rate of 20%. The histograms are the cross-validation values I both for the data (black) and its null reference (gray). In the case of k=5 (left), there is a clear distinction between the two histograms, whereas with k=8 there is no significant differences between the histograms.

Statistical analysis can now be performed to figure out in which k, the PDF of H_1 provides the strongest significant evidence against H_0 [9]. The basic approach is as follows. A significance level is set and I_1 and I_2 are determined based on the PDF of H_0. The number of datasets X for which $I>I_2$ are counted as p_1, where the total number of them is P. H_1 is accepted if p_1/P is larger than a threshold, for example 0.9.

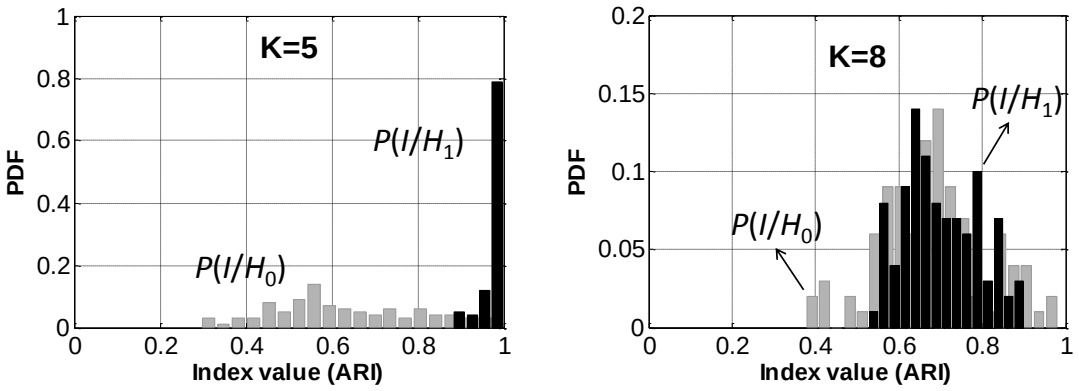


Figure 8. Null hypothesis testing for the dataset in Figure 1 for k=5 and k=8

2.5. Classification-based approach

The ideas from supervised learning have also been used to evaluate the stability of clustering results in terms of their reproducibility [9]. The data, in P independent iterations, is randomly divided into two disjoint sets, a training set X_i^{tr} and a test set X_i^{te} where i=1..P. Clustering is applied to both X_i^{tr} and X_i^{te} to produce partitions Y_i^{tr} and Y_i^{te}. Another partition $Y'_i{}^{te}$ is then predicted for X_i^{te} using a classifier trained on (X_i^{tr}, Y_i^{tr}) [9], [17]. The two partitions for the test set are compared using an external index, see Figure 9. The P index values corresponding to P test sets are then averaged. The process is repeated for the potential number of clusters in the range $k\in[k_{\min}, k_{\max}]$. The hypothesis is that the highest stability (the average similarity between the two partitions of the test set) is achieved for the correct number of clusters.

To derive the labels for the test dataset from the clustering of training dataset, a classifier such as linear discriminant analysis or K-nearest-neighbor is used for training [9], [15]. Selecting a good classifier is a challenging problem. In theory, the classifier is never optimal. A classifier can be derived based on the clustering process that has been used, which leads to a smaller error than that of a general classifier. For example, the nearest-neighbor classifier is suitable for single-link clustering, and the nearest centroid classifier is suitable for centroid-based clustering algorithms such as K-means [15]. In the case of model-based clustering, the labels can be directly determined from the model obtained in the training process without any a classifier [17].

The size of training and test sets should be selected carefully. In classification, usually a larger portion of data is considered for training. However, in the current problem, considering more data for training might be problematic. For example, in density-based clustering, different sizes of test and training sets result in different densities, which might lead to different clusterings. In this case, training and test sets should have the same size [6].

The normalization based on null reference as in (2) can also be used for the classification-based approach. Figure 10 shows the results of cross-validation and classification-based approaches with and without normalization for the dataset in Figure

1. The number of subsets is 100, each of the size 20% of the full dataset in the cross-validation approach. The percentages for the training and test sets in the classification-based approach are 80% and 20%, respectively. Random swap clustering algorithm [33] and *adjusted rand index* (ARI) [27] are used. The highest stability is found with k=5, the correct number of clusters.

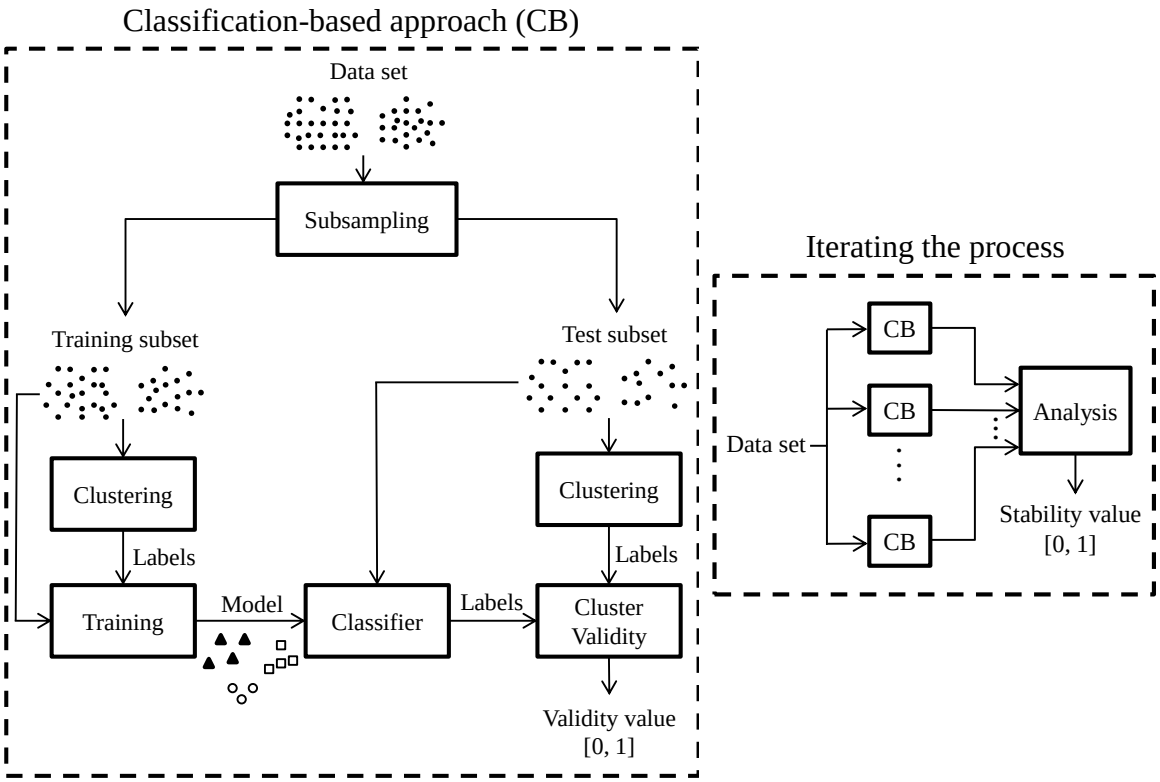


Figure 9. One iteration of the classification-based approach (left). The process is repeated by randomly generating several training and test sets (right).

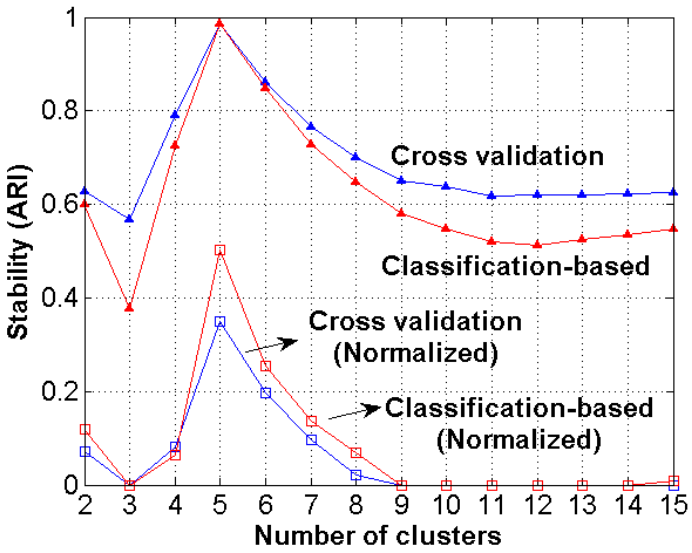


Figure 10. Example of the stability-based method for the dataset in Figure 1. The size of subsets in the cross-validation approach and test sets in the classification-based approach is 20%. Random swap algorithm is used for clustering [33] and *adjusted rand index* (ARI) for validation [27].

3. Clustering and validation

3.1. Clustering algorithm

A clustering algorithm should have two basic requirements to be suitable for the stability-based method. First, the algorithm itself should be stable so that it provides the same result when applied several times to the same data, or to several datasets drawn from the same source [15]. Otherwise, one cannot conclude whether the instability is caused by artifacts of the clustering algorithm, or by the structure of the data. Second, the algorithm must be suitable for the dataset so that it would be able to find a good solution for the correct number of clusters.

Existing clustering methods consider one of the following structures for the clusters:

1. Spherical
2. Gaussian
3. Density-based
4. Connectivity-based

K-means is the best known algorithm that aims at minimizing total squared error (TSE) [39]. It is suitable for spherical clusters but the result highly depends on the initial selection of centroids, and terminates in a local minimum. K-means is therefore unstable and not suitable for stability-based method. Random swap (RS) [33] is a more stable algorithm that iteratively changes the centroid locations through a trial-and-error manner. Due to its ease of implementation and stable performance, we use it as our baseline in the rest of this paper.

Another possible algorithm for spherical data is Agglomerative clustering (AC). Efficient implementation in [40] also minimizes TSE and it usually finds the correct clustering structure with minor inaccuracies near the cluster borders. It would be another suitable compromise of simplicity and stability. Best reported results for minimizing TSE has been obtained by Genetic algorithm (GA). The variant in [34] uses agglomerative clustering as genetic operations and k-means for fine-tuning the results. Other works in the stability-based literature for spherical clusters have used partitioning around medoids (PAM) [9], repeated k-means [4], [24], competitive learning [4], bisecting k-means [7], and average-link agglomerative clustering [16].

Gaussian data can be modeled by Gaussian mixture model. Most popular algorithm is expectation maximization (EM) [41], which is analogous to K-means. It iteratively improves the solution by a two-step process. It has been used in the stability-based method in [23]. The problem of EM is that it also depends on the initial configuration. Better algorithms include Split and merge EM (SMEM) [42], genetic algorithm EM [43], and random swap EM (RSEM) [44], which all aim at overcoming the problem of local optimum of EM. We expect them to provide more stable results than EM, and applicable to the stability-based methods.

Density-based clustering considers the clusters as areas of different densities, or higher density than the rest of data. Sparse points are usually considered as noise. DBSCAN [45] is the most popular density-based algorithm. Its basic idea is to create clusters from points whose neighborhood within a given radius (*eps*) contains a minimum number (*minPt*) of other points. The algorithm grows clusters from these points by joining neighboring points within the *eps* distance. The algorithm is simple but the result highly depends on the parameters *eps* and *minPt*. The number of clusters is also automatically selected based on these parameters, and is not applicable when there are clusters with

different densities in the data. OPTICS [46] generalizes DBSCAN so that the parameter *eps* is derived automatically. There are two main problems of these algorithms. First, they select the number of clusters *k* indirectly via the input parameters. Second, re-sampling the subsets will produce different densities than the original data, and therefore would need different parameters *eps* and *minPt*. These make it difficult to use the density-based algorithms in stability-based methods.

Connectivity-based clustering aims at forming arbitrary-shaped clusters by connecting nearby objects based on their distance. Agglomerative clustering with single-link and complete-link are two examples of connectivity-based clustering. Single-link would work only if the clusters are well-separated but provides poor results otherwise. Numerous algorithms appear in the literature but it is unclear which one of them would really work in practice.

In clustering, the main problem is that we usually do not know what kind of clusters is expected. However, we can still apply squared error criterion even when the clusters are not spherical, and find the clustering that fits for the assumed spherical clusters. It is therefore expected that we can still find the number of clusters that best fits the model for this data.

Figure 11 shows an example where the mismatch between the data and the clustering method can result in stability for an unstructured data. For example, an algorithm based on squared error criterion (k-means) is suitable for data with spherical clusters. When applying with k=2 to the spherical data without clusters, it provides unstable result (left), but when applying to skewed Gaussian data, it can provide stable result (right). Suppose that we have a data set with k clusters, k-1 spherical and one skewed as in Figure 11 (right). Applying the stability-based method with a centroid-based clustering to this data set provides stable result not only for k clusters but k+1 and k+2.

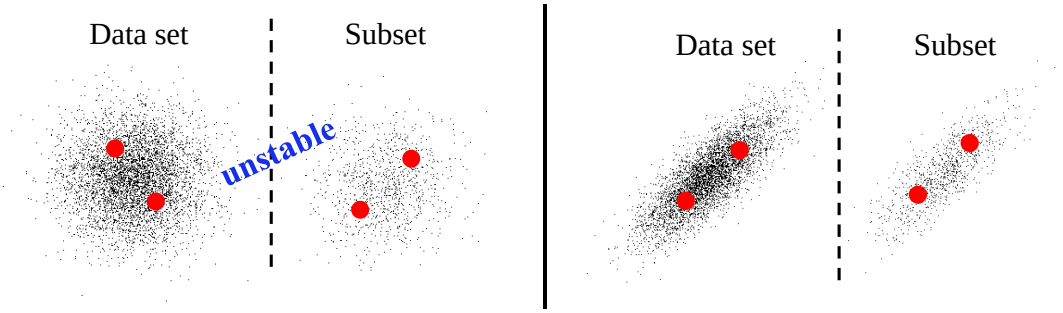


Figure 11. Datasets without structure; unstable clustering for a spherical 2-D dataset (left), and stable clustering for a skewed dataset (right) for k=2.

3.2. External validity index

External indices are categorized into *pair-counting*, *information theoretic* and *set-matching* measures [30]. Normalization and correction for chance are desirable properties. Normalization keeps the range of the index either in [-1, 1] or [0, 1], which makes the values comparable across different datasets. Correction for chance adjusts the expected value to zero [30].

Pair-counting measures include *rand index*, *adjusted rand index*, *Jaccard coefficient*, *Fowlkes-Mallows index* and several others [47], [48]. They count the pairs of objects in the dataset on which two different partitions agree or disagree. For instance, if two objects place in the same cluster, or in different clusters in the two clustering solutions,

it is an agreement. Rand index is defined as the number of agreements divided by the total number of pairs of objects. Adjusted rand index is the corrected form of Rand index for chance [27]:

$$ARI = \frac{RI - E(RI)}{1 - E(RI)} \tag{4}$$

where E(RI) is the expected value of Rand index. Adjusted rand index is the most popular index in this group.

Information theoretic indices include *entropy*, *mutual information* and *variation of information* [47], [29], [28]. Mutual information (MI) measures the information that two clusterings share by summing up the shared information between every two clusters:

$$MI(P,G) = \sum_{i=1}^{K}\sum_{j=1}^{K'} p(P_i, G_j) \log \frac{p(P_i, G_j)}{p(P_i)\,p(G_j)} \tag{5}$$

where the probabilities $p(P_i)$, $p(G_j)$, and $p(P_i,G_j)$ are estimated as n_i/N, n_j/N, and n_{ij}/N, respectively. N is the size of dataset, n_i and n_j are the sizes of clusters P_i and G_j, and n_{ij} is the number of shared objects between the two clusters. Variation of information (VI) represents the distance between two clusterings, and it is the complement of mutual information. Since there is no upper bound for mutual information and variation of information, normalization is needed [11]. Suppose that NVI_s=1- NVI and AVI_s=1-AVI denote the similarity form of normalized variation of information (NVI) and adjusted variation of information (AVI). It is shown in [30] that:

$$AVI_s = NVI_s = AMI = NMI \tag{6}$$

where NMI and AMI denote normalized mutual information and adjusted mutual information.

Set-matching indices are based on matching the clusters in two clustering solutions, where the similarity between every two clusters is calculated according to a given measure. We classify set-matching indices into two types: point-level such as *Van Dongen* [31] and *pair set index* [30], and cluster-level such as *Centroid Index* [25]. Cluster-level indices use only cluster prototypes in contrast to point-level indices, which employ the labels of all objects in resulting partitions. PSI and CI are defined as follows:

$$PSI = \begin{cases} \dfrac{S - E(S)}{\max(K,K') - E(S)} & S \geq E(S), \\ & \max(K,K') > 1 \\ 0 & S < E(S) \\ 1 & K = K' = 1 \end{cases}, \quad S = \sum_{i=1}^{\min(K,K')} \frac{n_{ij}}{\max(n_i, n_j)} \tag{7}$$

where i, j are the indices of paired clusters.

$$CI_1(P,G) = \sum_{i=1}^{K'} orphan(G_i) \tag{8}$$
$$CI_2(P,G) = \max(CI_1(P,G), CI_1(G,P))$$

where orphan(G_i)=1 if no centroid from clustering P is mapped to the ith centroid in clustering G, and zero otherwise. Since the mapping from P to G is not symmetric, CI_2 is defined by calculating the CI_1 measure in both ways.

Existing stability-based methods either define their own index or employ a well-known external index such as Fowlkes and Mallows (FM) [9], [24] and ARI [13] to measure the stability. However, the indices that they define are all similar to the existing external indices. For example, the index in [6] and [15] is a set matching-based index, which is corrected for chance. Optimal pairing for two partitions is derived and then the number of misclassified objects is calculated. The figure of merit [14] is a pair-counting external index, which counts the number of pairs of objects located in the same cluster in both clusterings. Prediction strength [49] is similar to the figure of merit, but the stability is measured only according to the cluster in the test set that has the minimum proportion of the pairs of objects.

4. Experiments

We examine the stability-based method for determining the number of clusters using several datasets. We arrange a set of experiments to answer the following questions:

- Which external index, and how to select *k*?
- Effect of sampling rate?
- Which Cross-validation strategy?
- Null reference or not?
- Which algorithm?

4.1. Experimental setup

External indices: We consider representative indices from every three categories of external indices: RI, ARI, NMI, CI, CSI, NVD and PSI. All the indices are traditional point-level normalized in the range [0, *k*] except CI that is a cluster level index in the range [0, *k*]. For visualization purpose, we normalize CI and convert it to a similarity measure in the range [0, 1] using CI*=1-CI/*k*. NVD is also a distance measure. We consider 1-NVD as a similarity measure in all the experiments.

Clustering algorithm: We use *random swap* (RS) [33] by-default in all tests unless otherwise noted. We set the number of its iterations to 5000 to make sure that the best possible clustering is achieved. K-means (KM) and genetic algorithm (GA) are used in one test to evaluate the impact of clustering algorithms.

Datasets: We consider 14 datasets summarized in Table 1. The datasets *S*1, *S*2, *S*3 and *S*4, each contains 15 clusters. The complexity and overlap between clusters increase from *S*1 to *S*4. *Iris* is a small dataset with three well-separated clusters. *Unbalance* is a dataset with three big clusters, each of size 1000 and five small clusters, each of size 100, see Figure 12. The points in the *Bridge* dataset are 4×4 non-overlapping vectors taken from a 256x256 gray-scale image. The dataset does not have any clustering structure. *Birch*1 and *Birch*2 are big datasets, each includes 100 well-separated spherical clusters. *G*2 is a series of datasets, all containing two clusters (1024 points each) but with varying dimension including 2-*d*, 4-*d*, 8-*d*, 16-*d*, and 32-*d*.

Subsampling: We generate 100 subsets from each dataset by random independent sub-sampling. The same subsets are used both in the cross-validation and as test sets in the classification-based approach in all experiments. The rest of the data (the complement of each subset) is used as training set in the classification-based approach. We generate 100 subsets similarly for the uniform null reference of each dataset. We consider sampling rates of 5%, 10%, 20%, 40% and 80%. By default, we use 20%.

Table 1. Summary of the datasets

Dataset	Number of data points (N)	Number of clusters (K)	Dimension of data (d)
*S*1- *S*4	5000	15	2
Iris	150	3	4
Unbalance	3500	8	2
Bridge	4096	-	16
*Birch*1	100,000	100	2
*Birch*2	100,000	100	2
*G*2	2048	2	2-4-8-16-32

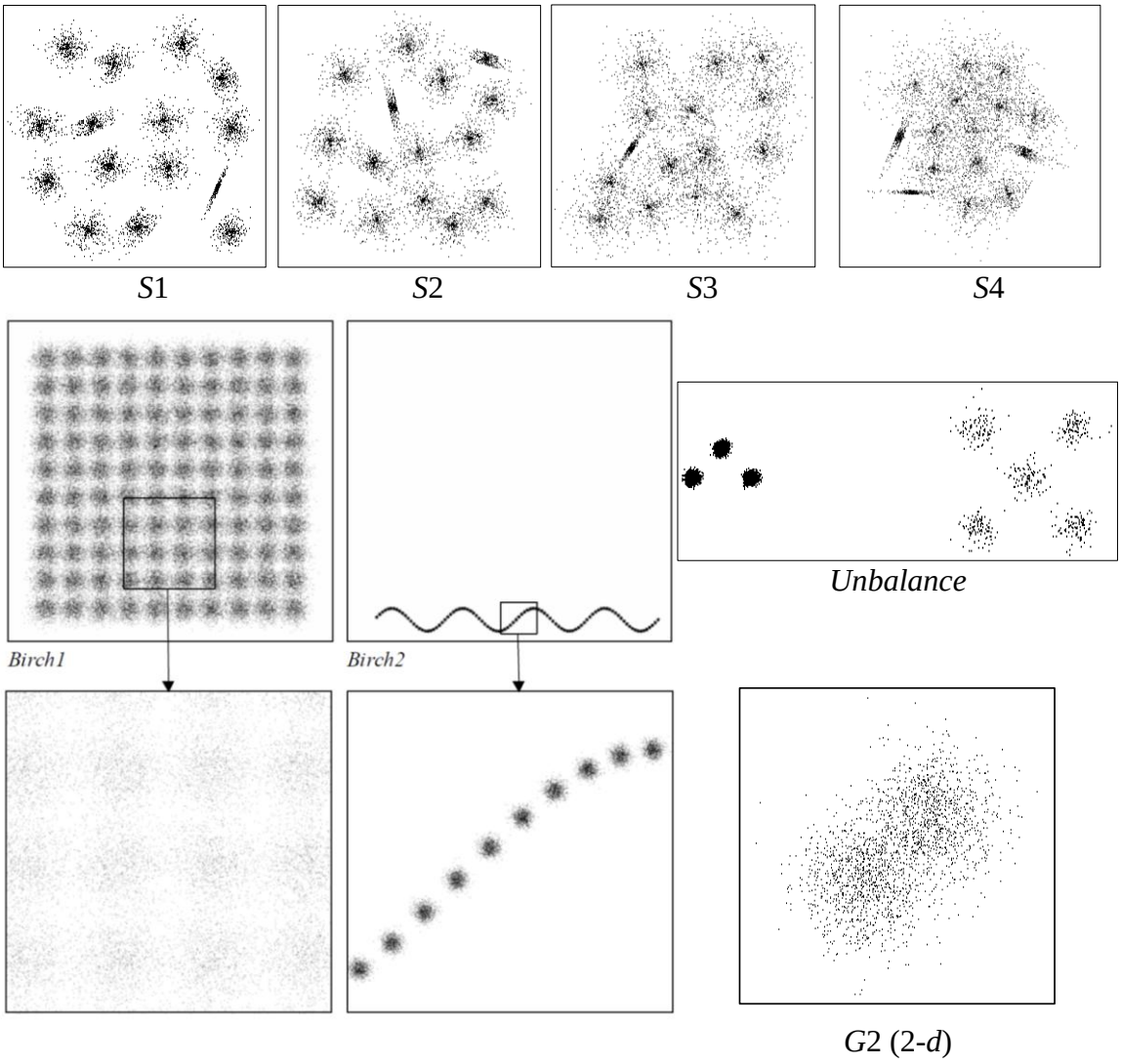


Figure 12. Example datasets

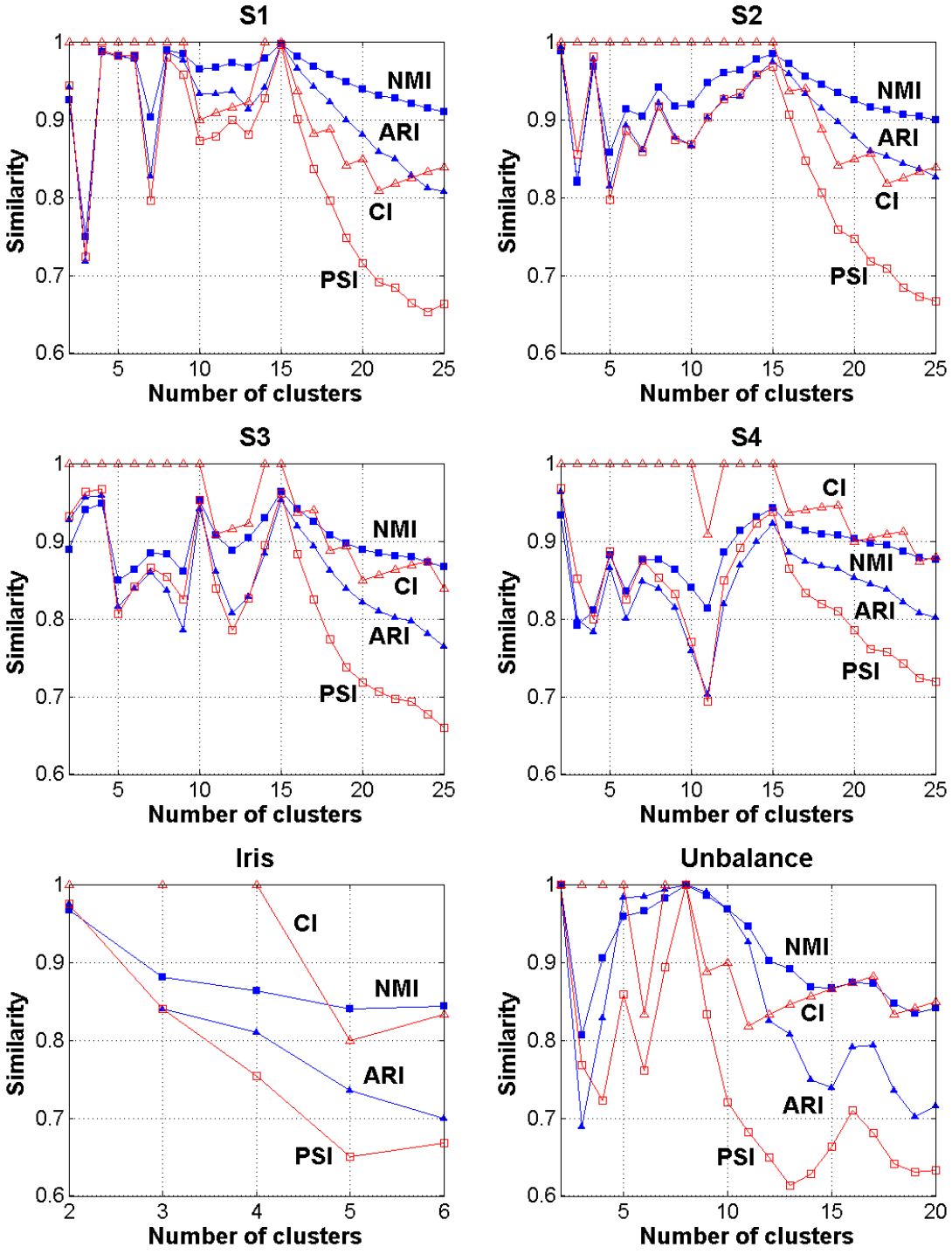
4.2. Comparison of external indices

We compare the performance of external indices using the subsampling-based cross-validation approach. We consider both the global maximum and the last local maximum approaches (with threshold 0.90).

Figure 13 shows the average index values for each dataset, and Table 2 records the number of clusters detected. First observation is that the choice of index is not important. Almost all the methods manage to find the correct result for most datasets.

Only exception is RI, which fail 43% times. The scale of CI is much rougher than the other indices. It always finds maximum stability for the correct number of clusters, but also several others when having too few clusters.

The second observation is that the global maximum criterion sometimes fails. It either detects multiple global maxima (especially with CI), or detects a solution with too few clusters. The last local maximum criterion works better in this sense. The following indices find the correct result in all cases except Iris: ARI, NMI, PSI, NVD, and CSI. CI measures only cluster level differences, and in some cases, it finds stability also when having too many clusters.



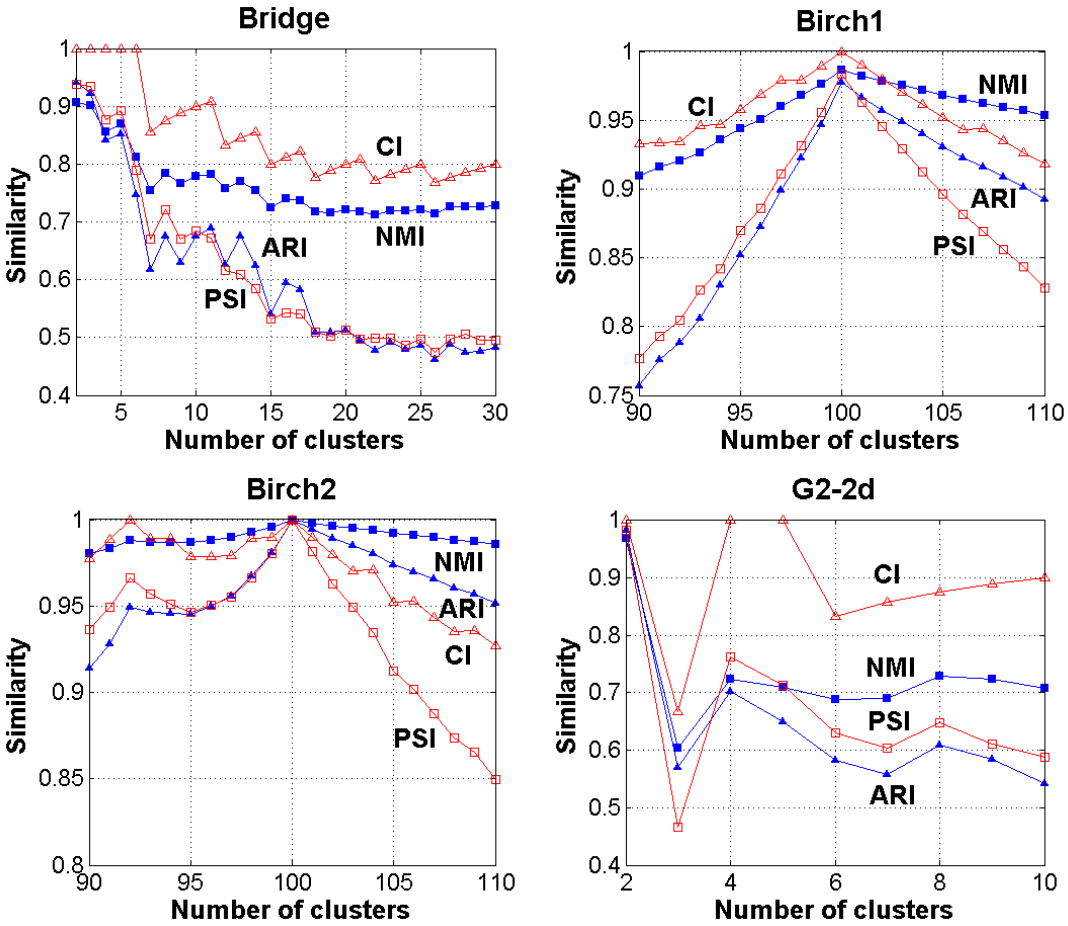


Figure 13. Stability results of cross-validation approach using various indices.

Table 2. Comparison of external indices when considering global maximum and last local maximum

	Datasets						
	Birch1	Birch2	G2-2d	G2-4d	G2-8d	G2-16d	G2-32d
	100	100	2	2	2	2	2
	Global maximum						
RI	99..105	100	2	2	2	2	2
ARI	100	100	2	2	2	2	2
NMI	100	100	2	2	2	2	2
PSI	100	100	2	2	2	2	2
NVD	100	100	2	2	2	2	2
CSI	100	100	2	2	2	2	2
CI	100	92, 100	2,4,5	2,3,4	2	2	2
	Last local maximum						
RI	100	100	9	2	2	2	2
ARI	100	100	2	2	2	2	2
NMI	100	100	2	2	2	2	2
PSI	100	100	2	2	2	2	2
NVD	100	100	2	2	2	2	2
CSI	100	100	2	2	2	2	2
CI	100	100	5	4	2	2	2

	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	3	8	1
	Global maximum						
RI	15	2, 15	15	15	2	2, 8	2
ARI	15	2	4	2	2	2, 8	2
NMI	15	2	15	15	2	2, 8	2
PSI	15	2	4	2	2	2, 8	2
NVD	15	2	3, 4	2	2	2, 8	2
CSI	15	2	3, 4	2	2	2, 8	2
CI	2..9,14,15	2..15	2..10, 14, 15	2..10, 12..15	2,3,4	2, 3, 4,5,7,8	2..6
	Last local maximum						
RI	15	15	23	19	2	17	27
ARI	15	15	15	15	2	8	2
NMI	15	15	15	15	2	8	2
PSI	15	15	15	15	2	8	2
NVD	15	15	15	15	2	8	5
CSI	15	15	15	15	2	8	5
CI	15	15	15	15	4	8	6

4.3. Cross-validation strategy

We compare next two cross-validation strategies with the classification-based approach using nearest centroid classifier. The results for two selected datasets are plotted in Figure 14. They show the same trend for both cross-validation (subset-fullset) and classification-based approaches with only slight differences.

To compare the clusterings of two subsets, we predict the labels of their full dataset using nearest centroid mapping. We then compare the resulting clusterings of the full dataset. Table 3 shows that there is no difference in the performance of the two cross-validation strategies and the classification-based approach when using last local maximum criterion. Global maximum criterion results in the same errors as in the previous experiment for all the approaches in this experiment.

We also tested randomization of the algorithm. Instead of k-means, we use more robust algorithm, random swap. We study the level of randomness by using 1, 10, 100, 1000 and 5000 iterations. Depending on the data, correct clustering is found by 10 (*G*2), 100 (*S*1-*S*4) or 5000 (*Birch*) iterations. Iterating less would cause more randomness potentially to allow detecting the number of clusters via stability. The results for the data sets *S1* and *Unbalance* (100 iterations) in Figure 14 shows the low performance of this approach. Both global maximum and last local maximum result in an incorrect number of clusters.

The results of the randomized algorithm in Table 3 show that it rarely works. Correct results are found only for the higher dimensional G2 datasets and with Birch, but only if the number of iterations is set properly: slightly less than what would be required to find

the optimum solution. Too few iterations would cause too much randomness, and stability will not be achieved even with the correct number of clusters. Too many iterations, on the other hand, allows the algorithm to find the same well-optimized clustering regardless of the initialization. Even with too many clusters, there is usually a unique global minimum that the algorithm finds. The fundamental problem of this approach is that the randomization cannot be controlled as easily as with the sub-sampling approach.

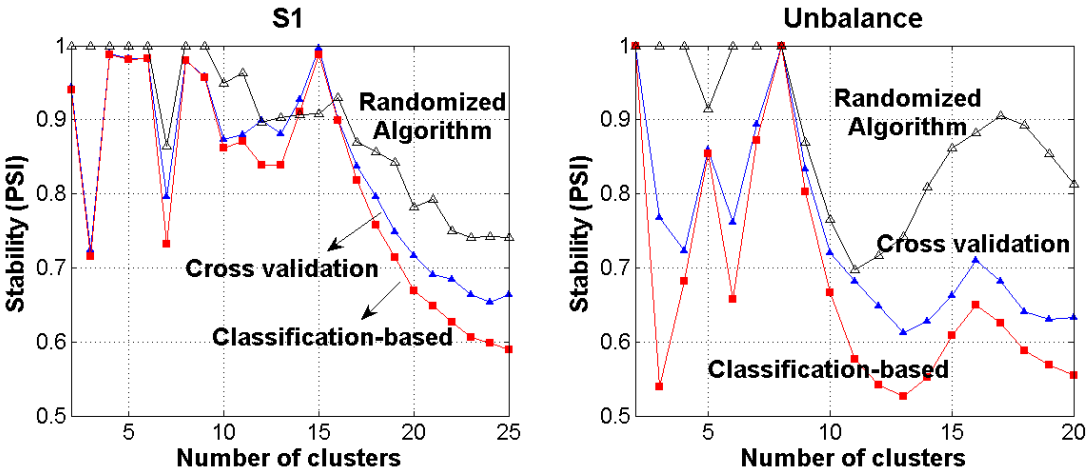


Figure 14. Comparison of cross-validation and classification-based approaches.

Table 3. Comparison of two cross-validation strategies vs. classification-based approach vs. randomized algorithm (R.A.) with different number of iterations

	Datasets						
	Birch1	Birch2	G2-2d	G2-4d	G2-8d	G2-16d	G2-32d
	100	100	2	2	2	2	2
Cross-valid. (FULL)	100	100	2	2	2	2	2
Cross-valid. (SUB)	100	100	2	2	2	2	2
Classification-based	100	100	2	2	2	2	2
R.A. (1)	1	1	2	2	2	2	2
R.A. (10)	1	1	4	2	2	2	2
R.A. (100)	98	1	8	2	2	2	2
R.A. (1000)	100	100	9	4	2	2	2
R.A. (5000)	100	109	9	3	2	2	2

	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	3	8	1
Cross-valid. (FULL)	15	15	15	15	2	8	2
Cross-valid. (SUB)	15	15	15	15	2	8	2
Classification-based	15	15	15	15	2	8	2
R.A. (1)	1	4	4	2	3	2	5
R.A. (10)	5	16	10	2	3	4	10
R.A. (100)	16	16	15	19	5	17	9
R.A. (1000)	19	16	22	22	6	17	10
R.A. (5000)	19	16	24	24	6	17	8

4.4. Null reference

We next test the normalization based on the null reference as used in (2). We report the results both using last local maximum and global maximum criteria with threshold 0.2. The results in Figure 15 reveals that the null reference is not monotonically decreasing as expected, but it fluctuates; only the magnitude of the fluctuation decreases with the number of clusters. The overall results (the difference) is affected badly by the fluctuation and leads to more confusion about the optimal number of clusters for most of the datasets, as shown in Table 4.

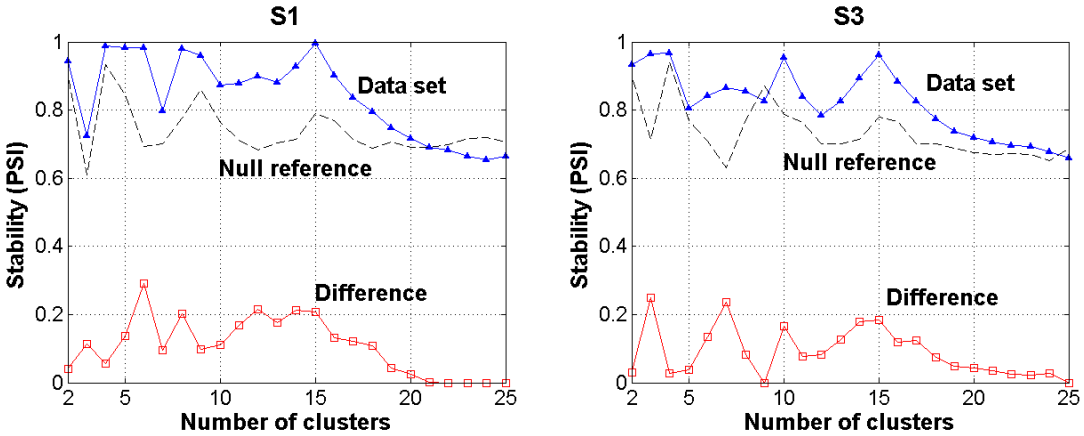


Figure 15. Comparing cross-validation approach without and with normalization using null reference

Table 4. Cross-validation approach without and with null reference: Global = global maximum, Local = last local maximum.

	Datasets						
	Birch1	Birch2	G2-2d	G2-4d	G2-8d	G2-16d	G2-32d
Without	100	100	2	2	2	2	2
Local	100	108	2	2	2	2	2
Global	100	100	2	2	2	2	2

	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
Without	15	15	15	15	2	8	2
Local	14	14	7	14	3	8	28
Global	6	7	3	14	3	8	3

4.5. Clustering algorithm

In this experiment, we compare the performance of three algorithms: random swap (RS), genetic algorithm (GA) and k-means (KM). The results for the datasets *S*1 and *Birch*1 in Figure 16 show that k-means results in lower stability values, which originates from the instability of the algorithm. This shows that the problem of evaluating the stability related to the structure of the data is mixed with the instability of the clustering algorithm. Therefore, wrong conclusions may be derived due to the choice of bad algorithm.

To determine the number of clusters, we use the last local maximum criterion with the thresholds 0.9, 0.9, and 0.7 for RS, GA, and k-means, respectively. The results in Table 5 confirm that k-means perform poorly for most datasets and is clearly unsuitable. The difference between RS and GA is so small that we expect some other algorithm like agglomerative might be suitable as well. We conclude that the choice of the algorithm is critical but several choices exist.

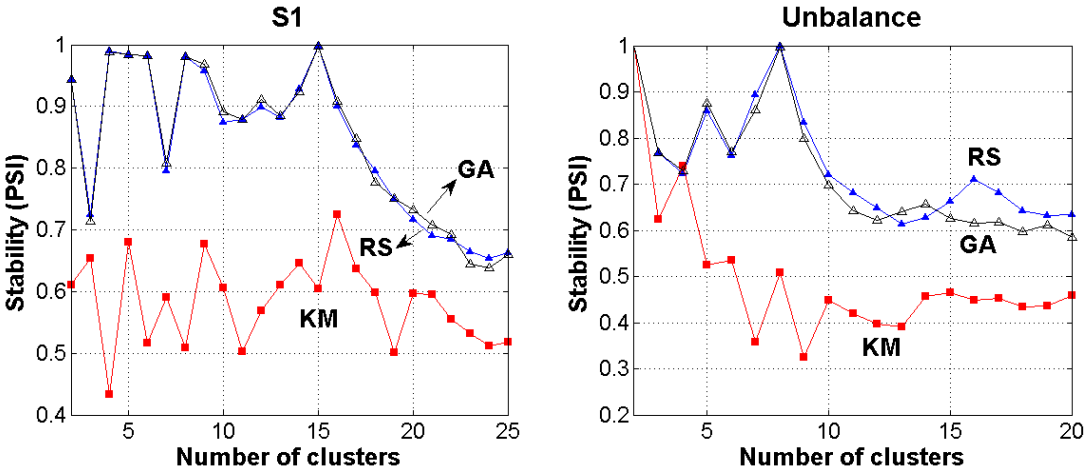


Figure 16. Comparison of k-means with two good algorithms: random swap and genetic algorithm

Table 5. Comparison of clustering algorithms by using PSI

Clustering algorithm	Datasets						
	Birch1	Birch2	G2-2d	G2-4d	G2-8d	G2-16d	G2-32d
RS	100	100	2	2	2	2	2
GA	100	100	2	2	2	2	2
KM	109	1	2	2	2	2	2

Clustering algorithm	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
RS	15	15	15	15	2	8	2
GA	15	15	15	15	2	8	2
KM	16	18	15	16	2	4	2

4.6. Impact of sampling rate

We test the impact of sampling rate on the performance of the cross-validation approach by generating subsets with several sampling rate including 5%, 10%, 20%, 40%, and 80%. Low sampling rate may cause too many changes in the structure of the data, whereas high sampling rate may result in too few changes. The results in Table 6 show that the subsampling rates 10%, 20% and 40% provide similarly good results. The low subsampling rate 5% causes error for S3 and S4, and the high sampling rate causes errors for S2, S4, and G2 (2-d).

Table 6. Impact of sampling rate on the cross-validation approach

Sampling rate	Datasets						
	Birch1	Birch2	G2-2d	G2-4d	G2-8d	G2-16d	G2-32d
5%	100	100	2	2	2	2	2
10%	100	100	2	2	2	2	2
20%	100	100	2	2	2	2	2
40%	100	100	2	2	2	2	2
80%	100	100	8	2	2	2	2

Sampling rate	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
5%	15	15	1	1	1	8	2
10%	15	15	15	15	2	8	2
20%	15	15	15	15	2	8	2
40%	15	15	15	15	2	8	5
80%	15	13	15	17	2	8	8

5. Conclusions

We have performed a systematic study to find out whether stability-based method can be used for determining the number of clusters. The simple answer is that, yes, it is possible but we think it is not practical. If it is going to be used, we give the following recommendations how to construct the method.

1. The exact choice of the cross-validation strategy is not critical. Most indices cannot compare the subsets directly, but comparing a subset to the full set works just fine. Random sub-sampling is suitable and no need to consider other approaches. The size of the subset is not a critical if between 10-40%. We recommend sub-sample size of 20%.

   Among the other cross-validation strategies: subset-subset and classification-based approach also work but they add an extra design question, predicting the labels for the missing points or the choice of classifier. These complicate the method unnecessarily. Randomizing algorithm is not recommended because it gives poor results even when using a stable algorithm.

2. To select the number of clusters, maximum stability criterion has been mostly used in the literature. However, we showed that it is not reliable. Instead, we recommend using the last local maximum criterion.

   Normalization based on the Null reference was also studied but it works poorly in most cases we tested. It does not bring enough extra insight into the process but adds more randomness that is harmful.

3. The choice of external index is not critical. Our results showed that any good external index (PSI, ARI, NMI, NVD, CSI) works and there is no significant difference between them. Only simple index like Rand Index had negative impact on the result and should not be used. Cluster level index (CI) was also considered because it works for the subset-subset comparison directly. It worked well but with two datasets it falsely recognized stability with too many clusters.

4. The choice of the clustering algorithm has significant effect on the result. The stability-based method fails using an unstable algorithm like k-means because it is simply not stable even with the correct number of clusters. We tested random swap (RS) and genetic algorithm (GA) but expect that other good algorithms such as agglomerative clustering (AC) would work well. We leave it as future research to study the stability of the algorithms more extensively.

Using the above guidelines, stability-based method can work with reasonable efforts. Despite the positive results, we encountered several challenges that might cause problems when applying the method in different contexts than what we studied here. We briefly discuss them next.

The method has two parameters to set: the sampling rate (20%) and the threshold of the last local maximum criterion (0.90). Too low (5%) or too high (80%) sampling rate makes the method fail for some data, and 40% was already a borderline case. The suitable range and recommended value 20% seems a safe choice but they are still parameters, and it is an open question how well they generalize to other types of data.

Another difficulty is that the clustering result does not depend only on the algorithm but the clustering model it uses. If we know that the data is spherical, then minimizing squared error by random swap clustering algorithm is OK. If we have Gaussian clusters, then we should use a good algorithm to optimize Gaussian mixture model. Random swap variant of EM [44] might work but it was not tested. For more complex data types such as density clustering, we do not even know which clustering model would work well enough. We expect that algorithms such as DBSCAN and single-link are likely to work poorly in general.

A bigger problem is that we do not usually know anything about the data, or it is a mix of different cluster types. However, the goal of clustering is simply to provide the best clustering for whatever model is chosen, including the number of clusters. In this scenario, the stability method can fail due to detecting stability with wrong number of clusters regarding both the model and the data.

To sum up, we conclude that external indices can be used for the problem but only in theory, and in very controlled environment when the type of data is well known and no surprises appear. In practice, this is rarely the case. Even if we demonstrated the method is working successfully for several datasets, we do not recommend it. External indices simply do not offer anything more that the best internal indices cannot offer, and they would just add unnecessary complications into the system.

References

1. Q. Zhao and P. Fränti, "WB-index: A sum-of-squares based index for cluster validity," *Data & Knowledge Engineering*, 92, pp. 77-89, 2014.
2. J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, 21(15), pp. 3201-3212, 2005.
3. G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, 50(2)**,** pp. 159-179, 1985.
4. E. Dimitriadou, S. Dolnicar, and A. Weingassel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika*, 67(1), pp. 137-159, 2002.
5. Q. Zhao, "*Cluster validity in clustering methods*," Phd. thesis, Dept. of Computer Science, University of Eastern Finland, 2012.
6. V. Roth, T. Lange, M. Braun, and J. Buhmann, "A resampling approach to cluster validation, " *Compstat*, pp. 123-128, Physica-Verlag HD, 2002.

7. E. Rendon, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. Journal of Computers and Communications*, 5(1), pp. 27-34, 2011.
8. M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intelligent Information Systems*, 17(2), pp. 107-145, 2001.
9. S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," *Genome biology*, 3(7), research0036, 2002.
10. L.I. Kuncheva and D.P. Vetrov, "Evaluation of stability of k-means cluster ensembles with respect to random initialization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11), pp. 1798-1808, 2006.
11. A. Strehl, J. Ghosh, and C. Cardie, "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," *J. Machine Learning Research*, 3, pp. 583-617, 2003.
12. S. Zhang, H. Wong and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognition*, 45(6), pp. 2214-2226, 2012.
13. Q. Zhao, M. Xu, and P. Franti, "Extending external validity measures for determining the number of clusters," *11th Int. Conf. Intelligent Systems Design and Applications (ISDA)*, pp. 931-936, 2011.
14. E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity," *Neural computation* 13(11), pp. 2573-2593, 2001.
15. T. Lange, V. Roth, M. Braun, and J. Buhmann. "Stability-based validation of clustering solutions," *Neural computation* 16(6), pp. 1299-1323, 2004
16. A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," *Pacific symposium on biocomputing*, 7, pp. 6-17, 2001.
17. J. N. Breckenridge, "Replicating cluster analysis: Method, consistency and validity," *Multivariate Behavioral research*, 24(2), pp.147-161, 1989.
18. U. V. Luxburg, "Clustering stability: An overview," *Foundations and Trends in Machine Learning*, 2(3), pp. 235-274, 2010.
19. U. Möller and D. Radke, "A cluster validity approach based on nearest-neighbor resampling," *18th Int. Conf. Pattern recognition*, 1, pp. 892-895, 2006.
20. M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, M. Hayward, and J. Trent., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, *406*(6795), pp. 536-540, 2000.
21. S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics*, 19(4), pp. 459-466, 2003.
22. J. Fridlyand and S. Dudoit, "Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method," *Tech. Report 600, Department of Statistics, UC Berkeley*, 31, 2001.
23. P. Smyth, "Clustering Using Monte Carlo Cross-Validation," *2nd Int. Conf. Knowledge Discovery and Data Mining*, pp. 126-133. 1996.
24. O. Abul, A. Lo, R. Alhajj, F. Polat, and K. Barker, "Cluster validity analysis using subsampling," *IEEE Int. Conf. Systems, Man and Cybernetics*, 2, pp. 1435-1440, 2003.
25. P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: cluster level similarity measure," *Pattern Recognition*, 47(9), pp. 3034-3045, 2014.
26. W.M. Rand, "Objective criteria for the evaluation of clustering methods," *J. American Statistical association*, *66*(336), pp. 846-850, 1971.
27. L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, pp. 193–218, 1985.
28. T.O. Kvalseth, "Entropy and correlation: some comments," *IEEE Trans. Syst. Man Cybern.*, 17(3), pp. 517–519, 1987.
29. M. Meila, "Comparing clusterings - an information based distance," *J. MultivariateAnalysis*, 98(5), pp. 873-895, 2007.
30. M. Rezaei and P. Fränti, "Set Matching Measures for External Cluster Validity," Manuscript, Manuscript (submitted).

31. S.V. Dongen, “Performance criteria for graph clustering and markov cluster experiments,” *Technical Report INSR0012*, Centrum voor Wiskunde en Informatica, 2000.
32. H. Frigui and R. Krishnapuram, "Clustering by competitive agglomeration", *Pattern Recognition*, 30 (7), 1109-1119, July 1997.
33. P. Fränti and J. Kivijärvi, “Randomised local search algorithm for the clustering problem,” *Pattern Analysis & Applications*, 3(4), pp. 358-369, 2000.
34. P. Fränti, "Genetic algorithm with deterministic crossover for vector quantization", *Pattern Recognition Letters*, 21 (1), 61-68, 2000.
35. P. Fränti, M. Rezaei, and Q. Zhao, “Centroid index: cluster level similarity measure,” *Pattern Recognition*, 47(9), pp. 3034-3045, 2014.
36. M. Halkidi, Y. Batistakis and M. Vazirgiannis, “Cluster validity methods: Part I,” *ACM SIGMOD Record*, 31(2), pp. 40-45, 2002.
37. R. Tibshirani, G. Walther, and T. Hastie, “Estimating the number of clusters in a data set via the gap statistic,” *J. the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2), pp. 411-423, 2001.
38. S. Theodoridis and K. Koutroumbas, “*pattern recognition*,” Fourth edition, Academic Press, 2008.
39. D. MacKay, “An example inference task: clustering,” *Information Theory, Inference and Learning Algorithms, Cambridge: Cambridge university press*, pp. 284-292, 2003.
40. P. Fränti, T. Kaukoranta, D.-F. Shen and K.-S. Chang, "Fast and memory efficient implementation of the exact PNN", *IEEE Trans. on Image Processing*, 9 (5), 773-777, May 2000.
41. A.P. Dempster, N.M. Laird, and D.B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. royal statistical society. Series B (methodological)*, pp. 1-38, 1977.
42. N. Ueda, R. Nakano, Z. Ghahramani, G.E. Hinton, “SMEM algorithm for mixture models,” *Neural computation*, *12*(9), pp. 2109-2128, 2000.
43. F. Pernkopf and D. Bouchaffra, “Genetic-based EM algorithm for learning Gaussian mixture models,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, *27*(8), pp. 1344-1348, 2005.
44. Q. Zhao, V. Hautamäki, I. Kärkkäinen, and P. Fränti, “Random swap EM algorithm for Gaussian mixture models,” *Pattern Recognition Letters*, *33*(16), pp. 2120-2126, 2012.
45. M. Ester, H. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” *Int. Conf. Knowledge discovery and data mining*, pp. 226-231, 1996.
46. M. Ankerst, M., M.M Breunig, H.P. Kriegel, and J. Sander, “OPTICS: ordering points to identify the clustering structure,” *ACM Sigmod Record*, 28(2), pp. 49-60. 1999.
47. N.X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance,” *J. Machine Learning Research*, 11, pp. 2837-2854, 2010.
48. A.N. Albatineh, M. Niewiadomska-Bugaj, and D. Mihalko, “On similarity indices and correction for chance agreement,” *J. Classification*, 23(2), pp. 301-313, 2006.
49. R. Tibshirani and G. Walther, “Cluster validation by prediction strength,” *J. Computational and Graphical Statistics*, 14(3), pp. 511-528, 2005.

Paper P4

Q. Zhao, M. Rezaei, and P. Fränti
"Keyword clustering for
automatic categorization",
International Conference on Pattern Recognition
(*ICPR*),
pp. 2845-2848, 2012.

21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Keyword Clustering for Automatic Categorization

Qinpei Zhao Mohammad Rezaei Hao Chen
Pasi Fränti
School of Computing, University of Eastern Finland
qinpei.zhao@uef.fi

Abstract

Processing short texts is becoming a trend in information retrieval. Since the text has rarely external information, it is more challenging than document. In this paper, keyword clustering is studied for automatic categorization. To obtain semantic similarity of the keywords, a broad-coverage lexical resource WordNet is employed. We introduce a semantic hierarchical clustering. For automatic keyword categorization, a validity index for determining the number of clusters is proposed. The minimum value of the index indicates the potentially appropriate categorization. We show the result in experiments, which indicates the index is effective.

1. Introduction

With the development on Internet, web-based and *information retrieval* (IR) applications, such as search engines, social networks, multi-media sharing, customer reviews are exploded. Short texts such as search query, comments, photo description and tags are the modern means in the applications. Although text classification and clustering are well studied, the techniques are not successful in dealing with short texts. The short text is typically lack of context information, free form and highly unstructured. Thus processing short texts is challenging. To enrich the representations of short texts, external resources such as WordNet [1], Wikipedia [2] and Google search results [1, 2, 4, 6, 7, 12] get involved.

Search engine queries are mostly short texts. The average length of them is about 2.3 terms and 30% have a single term [9]. A method grouping search results based on different meanings of the query is proposed in [4] for efficiently identifying relevant results. To get a better semantic similarity, search engine results are employed [12, 1, 2]. For each pair of short texts, they do statistics on the results returned by a search engine (e.g., Google) in order to get the similarity score.

New inspired clustering algorithms have been proposed to deal with short texts. In [5], a framework of comments-driven clustering for organizing web resources is explored. The clustering approach is studied over the popular video sharing site YouTube [3]. A probabilistic framework, which includes a knowledgebase (*Probase*) and certain inferencing techniques on top of the knowledgebase is proposed in [11]. The framework is to enable machines to perform human-like conceptualization. Experiments are conducted on conceptualizing textual terms and clustering short pieces of text such as Twitter [4] messages. Also novel uses of validity indexes have been presented in [3, 8]. An evaluation of different internal clustering validity indexes is presented to determine the possible correlation between the indexes and F-measure [8].

Hierarchical clustering commonly employed in text clustering, is a method of cluster analysis which seeks to build a hierarchy of clusters. It provides *dendrogram* as clustering results. Non-hierarchical procedures usually require the user to specify the number of clusters before any clustering and hierarchical methods routinely produce a series of solutions ranging from one cluster to n clusters (assume n objects in the data set). Numerous methods for determining the number of clusters have been proposed for numerical data [10]. However, there is little research on validity index for keyword clustering.

In this paper, a new validity index, which determines the number of clusters for semantic hierarchical clustering is proposed. The method is applied for automatic categorization. Our focus is on strings in a single word, based on which processing on strings in multiple words is also applicable. Since single words lack of content for statistical conclusion, we employ WordNet to get se-

[1] http://wordnet.princeton.edu
[2] http://www.wikipedia.org
[3] www.youtube.com
[4] https://twitter.com

mantic similarity directly. The main contribution of this paper is to introduce a new validity index in keyword clustering.

2. Semantic Hierarchical Clustering

Given a list of keywords $S = \{s_1, s_2, ..., s_n\}$, keyword categorization is to cluster them into groups, where the keywords in each group are semantically similar. The clusters are defined as $C = \{c_1, c_2, ..., c_k\}$. Hierarchical clustering can provide categorization with one to n clusters, i.e., $1 \leq k \leq n$.

A semantic hierarchical clustering requires a measure of semantic similarity between data. The similarity measure can be obtained from external resources and we use WordNet thesaurus in this paper. Information-content based similarity measures such as Resnik, Lin and Jiang & Conrath are considered. Take an example of Jiang & Conrath in distance metric, which is defined as:

$$\begin{aligned} P(s) &= \frac{\sum_{w \in Set(s,s')} count(w)}{N} \\ IC(s) &= -\log P(s) \\ LCS(s,s') &= \max_{c \in Set(s,s')} IC(c) \\ JC(s,s') &= (IC(s) + IC(s')) - 2LCS(s,s') \end{aligned} \tag{1}$$

where $Set(s, s')$ is a set of words subsumed by s and s'. $P(s)$ is the probability that a random word (w) in the corpus is an instance of s. N is the number of words in the corpus. $LCS(s, s')$ (Least Common subsumer) is the lowest common ancestor node of s and s' in the hierarchy of WordNet.

An example of semantic hierarchical clustering result by Jiang & Conrath is shown in Fig. 1.

3. Automatic Categorization

In most real life clustering situations, an applied researcher is faced with the dilemma of selecting the number of clusters in the final result. Thus, a validity index for determining the number of clusters is necessary. The index is based on the dendrogram with cluster size one to n obtained from hierarchical clustering (see Fig. 1). It is used to decide at which level of the hierarchy the categorization is the best.

For getting a proper number of clusters, a fixed range of $[k_{min}, k_{max}]$ is usually pre-defined. It is meaningless to set $k_{min} = 1$ because uniform test (deficiency of randomness) is enough. Also clustering algorithm has no effect on one cluster. Thus, usually one sets $k_{min} = 2$ and $k_{max} \leq n$.

The index is defined based on the *Compactness* and *Separation* of clusters, which are defined as:

$$\begin{aligned} C(k) &= \max_t \{\max_{i,j} JC(s_i, s_j)_{s_i \neq s_j \in c_t}\} + I_1/n \\ S(k) &= \frac{\sum_{t=1}^{k} \sum_{s>t}^{k} \min_{i,j} JC(s_i, s_j)_{s_i \in c_t, s_j \in c_s}}{k(k-1)/2} \end{aligned} \tag{2}$$

Where, $C(k)$ represents compactness within clusters and $S(k)$ is separation between clusters. In $C(k)$, s_i and s_j are the ith and jth string in tth cluster c_t and I_1 is the number of clusters with one item. Similarly, s_i and s_j are the ith and jth string in tth cluster c_t and sth cluster c_s respectively, k is the number of clusters at that hierarchical level.

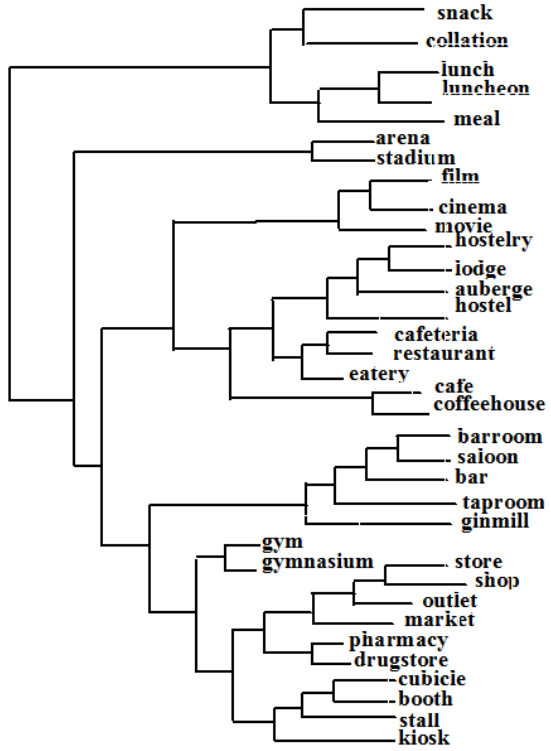


Figure 1. An example of dendrogram from semantic hierarchical clustering on data mopsi.

There exists a special case that cluster size is one, which means there is only one item in a cluster. For clustering, the special case is not preferred. And it is not possible to calculate the pairwise distance with only one item. Thus, we constraint the $C(k)$ by adding I_1/n. The categorization is assumed that items within a cluster are as similar as possible and items between clusters

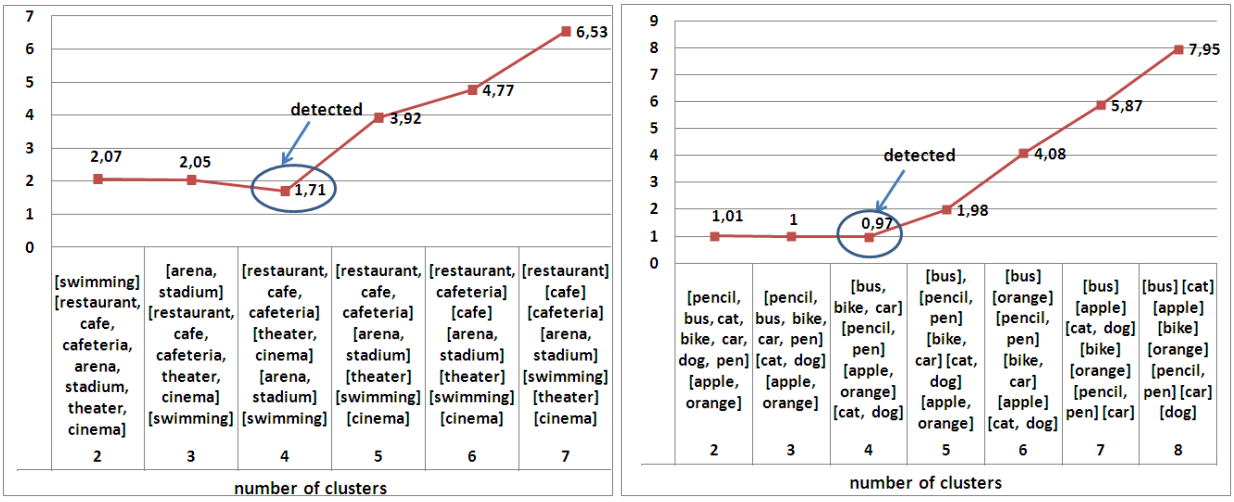


Figure 2. The stopping criterion on artificial data. Four is the minimum value for both cases.

are as different as possible. For the clustering result with k clusters, the validity index is defined as:

$$SC(k) = \frac{C(k)}{S(k)} \quad (3)$$

The index is calculated for each k among $[k_{min}, k_{max}]$. The k with minimum value in the range is selected as the best fitting number of clusters.

4. Experiments

The experiment is conducted on artificial data (see Fig. 2) and data mopsi obtained from *MOPSI*[5] project. The MOPSI project implements different location-based services and applications such as mobile search engines, photos, user tracking and route recording. The project has its applications integrated both on the web and mobile phones with the aim to integrate user location as a search option. The words in data mopsi (see Fig. 1), which contains 36 nouns, are picked up from services, search query keywords and photo descriptions. Since there are many unstructured words in Finnish language, we select a small sample and translate them into English by Google Translate API. We use Java to access the semantic similarity measures from WordNet 3.0. The user interface is programmed in JSP (Java Server Pages).

The validity index on artificial data is shown in Fig. 2. The x-axis is the number of clusters k and y-axis is the value of $SC(k)$. The categorizations are also displayed. The numbers of clusters detected by the stopping criterion are both four, where the categorization is reasonable from human judgment.

[5] http://cs.joensuu.fi/mopsi

For the real data mopsi, a ground truth categorization is obtained by 20 people. There are two persons who divide the data into 11 clusters, five persons into 10 clusters and 13 persons into 8 clusters. The dendrogram by the semantic hierarchical clustering is shown in Fig. 1. The number of clusters detected by the proposed validity index is nine (see Fig. 3), where the values of seven, eight and ten clusters are quite close. The categorization of nine groups is shown in Fig. 4. The maximum distances ($JC(s, s')$) within clusters and the minimum distances between clusters are displayed.

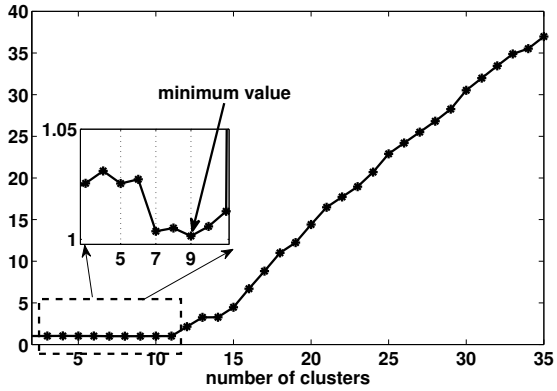


Figure 3. The validity index on data mopsi. Nine is the minimum value.

Although a number of clusters can be determined by an algorithm based on a certain criterion, human judgment often differs from each other on the categorizations and the number of clusters. However, the proposed criterion can suggest a potentially appropriate

categorization.

The study is simply performed on nouns. It can be extended to verbs also. For strings with multiple words, the processing can be based on processing for strings in single words. However, it is more complicated to analyze strings in multiple words by WordNet.

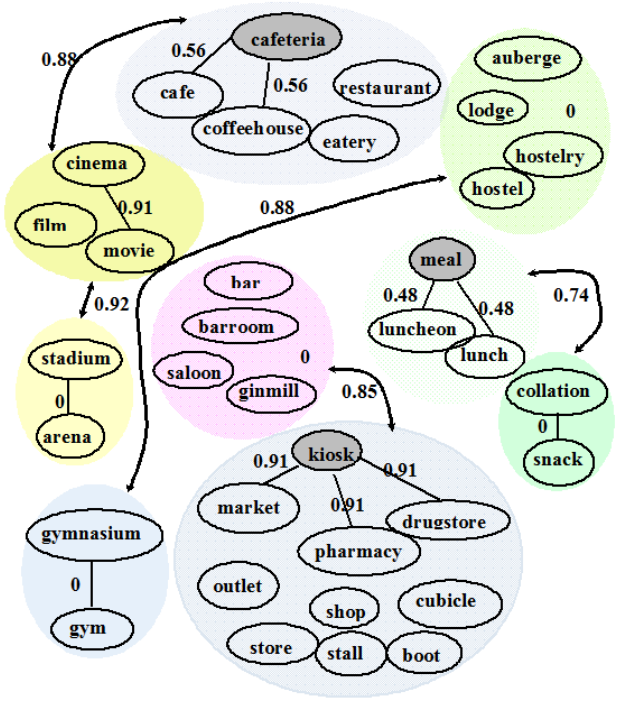


Figure 4. Categorization of nine groups on data mopsi with the minimum distances within clusters and maximum distances between clusters.

The semantic similarity obtained from WordNet sometimes has difference with human's judgment, which leads to the undesired clustering result. For example, the similarity between words *lion* and *tomcat* is 0, however, the similarity between *lion* and *cancer* is 0.05. The hierarchical clustering merges *lion* and *cancer* as a group firstly, which does not match with human's judgment. Therefore, automatic categorization on the undesired clustering result is not reliable.

5. Conclusion

We introduced a keyword clustering for automatic categorization. For getting a semantic similarity, we employed the similarity measure from WordNet. A validity index in semantic hierarchical clustering was proposed for automatic categorization. The index is based on the compactness and separation of clusters, where the minimum value indicates a good categorization. The experiment performed in a real project indicates the method is working. Finding a better way to calculate semantic similarity for strings in either single word or multiple words is our future work. It is also interesting to study on other clustering algorithms, such as spectral clustering on this problem.

References

[1] D. Bollegala, Y. Matsuo, and M. Ishizuka. Measuring semantic similarity between words using web search engines. *Proc. WWW*, pages 757–766, 2007.

[2] R. Cilibrasi. The google similarity distance. *IEEE Trans. on Knowledge and Data Engineering*, 19(3):370–383, 2007.

[3] M. Errecalde, D. Ingaramo, and P. Rosso. A new anttree-based algorithm for clustering short-text corpora. *Journal of Computer Science and Technology*, 10(1):1–7, 2010.

[4] R. Hemayati, W. Meng, and C. Yu. Semantic-based grouping of search engine results using wordnet. *APWeb/WAIM'07*, pages 678–686, 2007.

[5] C. Hsu, J. Caverlee, and E. Khabiri. Hierarchical comments-based clustering. *Proc. of the 2011 ACM Symposium on Applied Computing*, pages 1130–1137, 2011.

[6] J. Hu, L. Fang, Y. Cao, H. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging wikipedia semantics. *Proc. of the 31st annual intl. ACM SIGIR conf. on Research and development in information retrieval*, pages 179–186, 2008.

[7] X. Hu, N. Sun, C. Zhang, and T. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. *CIKM'09*, pages 919–928, 2009.

[8] D. Ingaramo, D. Pinto, P. Rosso, and M. Errecalde. Evaluation of internal validity measures in short-text corpora. *CICLing'08*, pages 555–567, 2008.

[9] B. Jansen, B. Spink, J. Bateman, and T. Saraceric. Real life information retrieval: a study of user queries of the web. *ACM SIGIR Forum*, 32(1):5–17, 1998.

[10] G. Milligan and M. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 50:159–179, 1985.

[11] Y. Song, H. Wang, Z. Wang, and H. Li. Short text conceptualization using a probabilistic knowledgebase. *TechReport:MSR-TR-2011-26*, 2011.

[12] W. Yih and C. Meek. Improving similarity measures for short segments of text. *Proc. AAAI*, 2:1489–1494, 2007.

Paper P5

M. Rezaei and P. Fränti,
"Matching similarity for
keyword-based clustering",
Joint IAPR International Workshop, S+SSPR,
pp. 193-202, 2014.

Matching Similarity for Keyword-Based Clustering

Mohammad Rezaei and Pasi Fränti

University of Eastern Finland
{rezaei,franti}@cs.uef.fi

Abstract. Semantic clustering of objects such as documents, web sites and movies based on their keywords is a challenging problem. This requires a similarity measure between two sets of keywords. We present a new measure based on matching the words of two groups assuming that a similarity measure between two individual words is available. The proposed matching similarity measure avoids the problems of traditional measures including minimum, maximum and average similarities. We demonstrate that it provides better clustering than other measures in a location-based service application.

Keywords: clustering, keyword, semantic, hierarchical.

1 Introduction

Clustering has been extensively studied for text mining. Applications include customer segmentation, classification, collaborative filtering, visualization, document organization and indexing. Traditional clustering methods consider numerical and categorical data [1], but recent approaches consider also different text objects such as documents, short texts (e.g. topics and queries), phrases and terms.

Keyword-based clustering aims at grouping objects that are described by a set of *keywords* or *tags*. These include movies, services, web sites and text documents in general. We assume here that the only information available about each data object is its keywords. The keywords can be assigned manually or extracted automatically. Fig. 1 shows an example of services in a location-based application where the objects are defined by a set of keywords. For presenting an overview of available services to a user in a given area, clustering is needed.

Several methods have been proposed for the problem [2, 3, 4, 5] mostly by agglomerative clustering based on single, compete or average links. The problem is closely related to *word clustering* [6, 7, 8] but instead of single words, we have a set of words to be clustered. Both problems are based on measuring similarity between words as the basic component.

To solve clustering, we need to define a similarity (or distance) between the objects. In agglomerative methods such as *single link* and *complete link*, similarity between individual objects is sufficient, but in partitional clustering such as *k-means* and *k-medoids* cluster representative is also required to measure object-to-cluster similarity. Using semantic content, however, defining the representative of a cluster is not trivial. Fortunately, it is still possible to apply partitional clustering even without the representatives. For example, an object can be assigned to such cluster that minimizes

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 193–202, 2014.

Fig. 1. Five examples of location-based services in Mopsi (http://www.uef.fi/mopsi): name of the service, representative image, and the keywords describing the service

(or maximizes) the cost function where only the similarities between objects are needed.

In this paper, we present a novel similarity measure between two sets of words, called *matching similarity*. We apply it to keyword-based clustering of services in a location-based application. Assuming that we have a measure for comparing semantic similarity between two words, the problem is to find a good measure to compare the sets of words. The proposed matching similarity solves the problem as follows. It iteratively pairs two most similar words between the objects and then repeats the process for the rest of the objects until one of the objects runs out of words. The remaining words are then matched just to their most similar counterpart in the other object.

The rest of the paper is organized as follows. In Section 2, we review existing methods for comparing the similarity of two words, and select the most suitable for our need. The new similarity measure is then introduced in Section 2. It is applied to agglomerative clustering in Section 3 with real data and compared against existing similarity measures in this context.

2 Semantic Similarity between Word Groups

In this section, we first review the existing methods for measuring semantic similarity between individual words, because it is the basic requirement for comparing two sets of words. We then study how they can be used for comparing two set of words, present the new measure called *matching similarity*, and demonstrate how it is applied in clustering of services in a location based application.

2.1 Similarity of Words

Measures for semantic similarity of words can be categorized to *corpus-based*, *search engine-based, knowledge-based* and *hybrid*. Corpus-based measures such as *point-wise mutual information* (PMI) [9] and *latent semantic analysis* (LSA) [9] define the similarity based on large corpora and term co-occurrence. Search engine-based measures such as *Google distance* are based on web counts and snippets from results of a search engine [8], [10, 11]. *Flickr distance* first searches two target words separately through the image tags and then uses image contents to calculate the distance between the two words [12].

Knowledge-based measures use lexical databases such as *WordNet* [13] and *CYC* [13], which can be considered as computational format of large amounts of human knowledge. The knowledge extraction process is very time consuming and the database depends on human judgment and it does not scale easily to new words, fields and languages [14, 15].

WordNet is a taxonomy that requires a procedure to derive the similarity score between words. Despite its limitations it has been successively used for clustering [16]. Fig. 2 illustrates a small part of WordNet hierarchy where mammal is the *least common subsumer* of wolf and hunting dog. *Depth* of a word is the number of links between it and the root word in WordNet. As an example, Wu and Palmer measure [17, 18] is defined as follows:

$$S(w_1, w_2) = \frac{2 \times depth(LCS(w_1, w_2))}{depth(w_1) + depth(w_2)} \tag{1}$$

where *LCS* is the least common subsumer of the words w_1 and w_2.

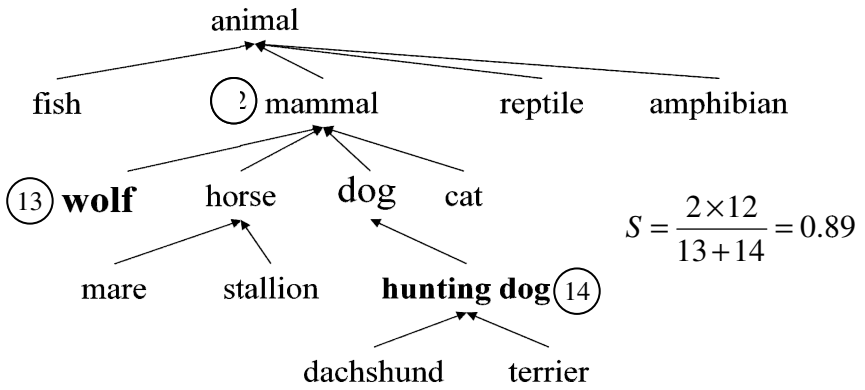


Fig. 2. Part of WordNet taxonomy; the numbers in the circles represent the depths

Jiang-Contrath [13] is a hybrid of corpus-based and knowledge-based as it extracts the information content of two words and their LCS in a corpus. Methods based on Wikipedia or similar websites are also hybrid in the sense that they use organized corpora with links between documents [19]. In the rest of the paper, we use Wu & Palmer measure due to its simplicity and reasonable results in earlier work [16].

2.2 Similarity of Word Groups

Given a measure for comparing two words, our task is to measure similarity between two sets of words. Existing measures calculate either minimum, maximum or average similarities. Minimum and maximum measures find the pair of words (one from each object) that are least (minimum) and most (maximum) similar. Average similarity considers all pairs of words and calculates their average value. Example is shown in Fig. 3, where the values are min=0.21, max=0.84, average=0.57.

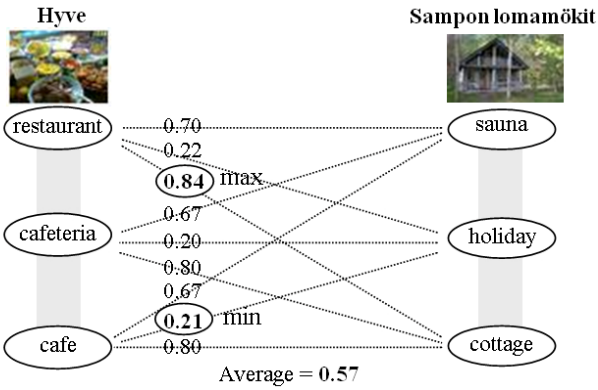


Fig. 3. Minimum and maximum similarities between two location-based services is derived by considering two keywords with minimum and maximum similarities

Now consider two objects with exactly the same keywords (100% similar) as follows:

(a) Café, lunch
(b) Café, lunch

The word similarity between Café and lunch is 0.32. The corresponding minimum, average and maximum similarity measures would result in 0.32, 0.66 and 1.00. It is therefore likely that minimum and average measures would cluster these in different groups and only maximum similarity would cluster them correctly in the same group.

Now consider the following two objects that have a common word:

(a) Book, store
(b) Cloth, store

The maximum similarity measure gives 1.00 and therefore as soon as the agglomerative algorithm processes to these objects, it clusters them in one group. However, if data contains lots of stores, they might have to be clustered differently.

The following example reveals another disadvantage of minimum similarity. These two objects should have a high similarity as their only difference is the drive-in possibility of the first service.

(a) Restaurant, lunch, pizza, kebab, café, drive-in
(b) Restaurant, lunch, pizza, kebab, café

Minimum similarity would result to S(drive-in, pizza)=0.03, and therefore, place the two services in different clusters.

2.3 Matching Similarity

The proposed *matching similarity* measure is based on a greedy pairing algorithm, which first finds two most similar words across the sets, and then iteratively matches next similar words. Finally, the remaining non-paired keywords (of the object with more keywords) are just matched with the most similar words in the other object. Fig. 4 illustrates the matching process between two sample objects.

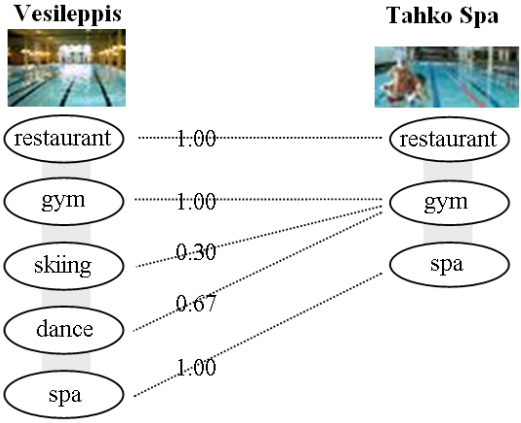


Fig. 4. Matching between the words of two objects

Consider two objects with N_1 and N_2 keywords so that $N_1 > N_2$. We define the normalized similarity between the two objects as follows:

$$S(O_1, O_2) = \frac{\sum_{i=1}^{N_1} SW(w_i^{O_1}, w_{p(i)}^{O_2})}{N_1} \quad (2)$$

where SW measures the similarity between two words, and $p(i)$ provides the index of the matched word for w_i in the other object.

The proposed measure provides more intuitive results than existing measures, and eliminates some of their disadvantages. As a straightforward property it gives the similarity 1.00 for the case of objects with same set of keywords.

3 Experiments

We study the method with Mopsi data (http://www.uef.fi/mopsi), which includes various location-tagged data such as services, photos and routes. Each service includes a set of keywords to describe what it has to offer. Both English and Finnish languages keywords have been casually used. For simplicity, we translated all Finnish words into English by Microsoft Bing translator for these experiments. Some issues raised in translation such as stop words, Finnish word converting to multiple English words, and some strange translations due to using automatic translator. We manually refined the data to remove the problematic words and the stop words.

In total, 378 services were used for evaluating the proposed measure and compare it against the following existing measures: *minimum*, *maximum* and *average similarity*. We apply complete and average link clustering algorithms as they have been widely used in different applications. Each of the clustering algorithms is performed based on three similarity measures. Here we fixed the number of clusters to 5 since our goal of clustering is to present user the main categories of services, with easy navigation to find the desired target without going through a long list. We find the natural number

of clusters using *SC* criteria introduced in [16] by finding minimum *SC* value among clusterings with different number of clusters. We then display four largest clusters and put all the rest in the fifth cluster. The data and the corresponding clustering results can be found here (http://cs.uef.fi/paikka/rezaei/keywords/).

The three similarity measures of five selected services in Table 1 are demonstrated in Table 2. The first three and the last two services should be in two different clusters according to their similarities. However, both minimum and average similarities show small differences when they compare *Parturi-kampaamo Nona* with *Parturi-kampaamo Koivunoro* and *Kahvila Pikantti*, whereas the proposed matching similarity can differentiate them much better. Despite that *Parturi-kampaamo Nona* and *Parturi-kampaamo Koivunoro* have exactly the same keywords, only the matching similarity provides value 1.00 indicating perfect match.

Table 1. Similarities between five services for the measures: minimum, average and matching

Mopsi service:	A1-Parturi-kampaamo Nona	A2-Parturi-kampaamo Platina	A3-Parturi-kampaamo Koivunoro	B1-Kielo	B2-Kahvila Pikantti
Keywords;	barber hair salon	barber hair salon	barber hair salon shop	cafe cafeteria coffe lunch	lunch restaurant

Table 2. Similarity between services described in Table 1

Services	A1	A2	A3	B1	B2
	Minimum similarity				
A1	-	0.42	0.42	0.30	0.30
A2	0.42	-	0.42	0.30	0.30
A3	0.42	0.42	-	0.30	0.30
B1	0.30	0.30	0.30	-	0.32
B2	0.30	0.30	0.30	0.32	-
	Average similarity				
A1	-	0.67	0.67	0.47	0.51
A2	0.67	-	0.67	0.47	0.51
A3	0.67	0.67	-	0.48	0.51
B1	0.47	0.47	0.48	-	0.63
B2	0.51	0.51	0.51	0.63	-
	Matching similarity				
A1	-	1.00	0.99	0.57	0.56
A2	1.00	-	0.99	0.57	0.56
A3	0.99	0.99	-	0.55	0.56
B1	0.57	0.57	0.55	-	0.90
B2	0.56	0.56	0.56	0.90	-

In general, the problems of minimum and average similarities are observable in the clustering results both for complete and average link. Several services with the same set of keywords (barber, hair, salon) are clustered together, and a service with the same keywords has its own cluster when complete link clustering is applied with minimum similarity measure. Average link method clusters the services with these keywords correctly but for services with other keywords (sauna, holiday, cottage), it clusters them in different groups even when using average similarity. This problem does not happen with matching similarity.

Another observation of minimum similarity with complete link clustering is that there appear many clusters with only one object, and a very large cluster that contains most of the other objects. Matching similarity leads to more balanced clusters with both algorithms. Interestingly, it also produces almost the same clusters with the two different clustering methods.

For more extensive objective testing, we should have a ground truth for the wanted clustering but this is not currently available as it is non-trivial to construct. We therefore make indirect comparison by using the *SC* criterion from [16]. The assumption here is that the smaller the value, the better is the clustering. Fig. 5 summarizes the SC-values for different number of clusters. The overall minima for complete link and average link are 131, 86, 146 (minimum, average and matching similarities) and 279, 96 and 140, respectively. Our method provides always the minimum *SC* value. The sizes of 4 biggest clusters in each case are listed in Table 3.

Table 3. The sizes of the four largest clusters for complete and average link clustering

Complete link				
Similarity:	Sizes of 4 biggest clusters			
Minimum	106	88	18	18
Average	44	22	20	19
Matching	27	23	19	17
Average link				
Similarity:	Sizes of 4 biggest clusters			
Minimum	22	12	10	8
Average	128	41	34	17
Matching	27	23	17	17

The effectiveness of the proposed method for displaying data with limited number of clusters still exists. The number of clusters is too large for practical use and we need to improve the clustering validity index to find larger clusters but without creating meaningless clusters. We also observed some issues in clustering that originate from the similarity measure of two words, which implies that better similarity measure would also be useful.

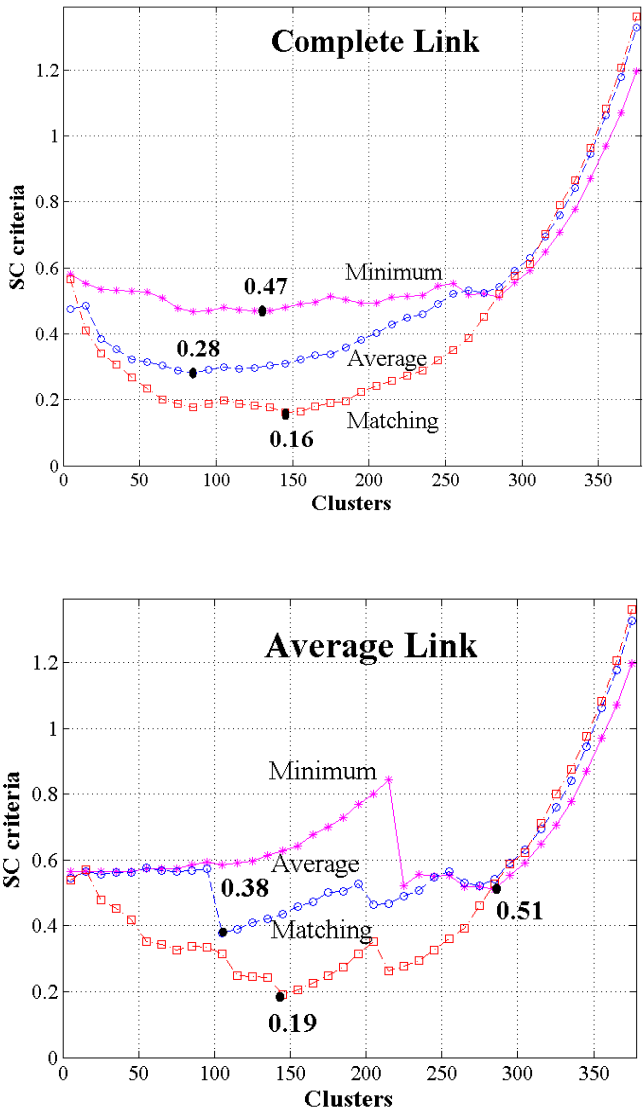


Fig. 5. Complete link and average link clustering using three similarity measures

4 Conclusion

A new measure called matching similarity was proposed for comparing two groups of words. It has simple intuitive logic and it avoids the problems of the considered minimum, maximum and average similarity measures, which fail to give proper results with rather simple cases. Comparative evaluation on a real data with SC criterion

demonstrates that the method outperforms the existing methods in all cases, and by a clear marginal. A limitation of the method is that it depends on the semantic similarity measure between two words. As future work, we plan to generalize the matching similarity to other clustering algorithms such as k-means and k-medoids.

Acknowledgements. This research has been supported by MOPIS project and partially by Nokia Foundation grant.

References

1. Aggarwal, C.C., Zhai, C.: A survey of text clustering algorithms. In: Mining Text Data, pp. 77–128. Springer US (2012)
2. Ricca, F., Pianta, E., Tonella, P., Girardi, C.: Improving Web site understanding with keyword-based clustering. Journal of Software Maintenance and Evolution: Research and Practice 20(1), 1–29 (2008)
3. Hasan, B., Korukoglu, S.: Analysis and Clustering of Movie Genres. Journal of Computing 3(10) (2011)
4. Ricca, F., Tonella, P., Girardi, C., Pianta, E.: An empirical study on keyword-based web site clustering. In: Proceedings of the 12th IEEE International Workshop on Program Comprehension. IEEE (2004)
5. Kang, S.S.: Keyword-based document clustering. In: Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, vol. 11. Association for Computational Linguistics (2003)
6. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of English words. In: Proceedings of the 31st Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics (1993)
7. Ushioda, A., Kawasaki, J.: Hierarchical clustering of words and application to NLP tasks. In: Proceedings of the Fourth Workshop on Very Large Corpora (1996)
8. Matsuo, Y., Sakaki, T., Uchiyama, K., Ishizuka, M.: Graph-based word clustering using a web search engine. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (2006)
9. Mihalcea, R., Corley, C., Strapparava, C.: Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, vol. 6 (2006)
10. Cilibrasi, R.L., Vitanyi, P.: The google similarity distance. IEEE Transactions on Knowledge and Data Engineering 19(3), 370–383 (2007)
11. Bollegala, D., Matsuo, Y., Ishizuka, M.: A web search engine-based approach to measure semantic similarity between words. IEEE Transactions on Knowledge and Data Engineering 23(7), 977–990 (2011)
12. Wu, L., et al.: Flickr distance: a relationship measure for visual concepts. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(5), 863–875 (2012)
13. Budanitsky, A., Hirst, G.: Evaluating wordnet-based measures of lexical semantic relatedness. Computational Linguistics 32(1), 13–47 (2006)
14. Kaur, I., Hornof, A.J.: A comparison of LSA, WordNet and PMI-IR for predicting user click behavior. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM (2005)

15. Gledson, A., Keane, J.: Using web-search results to measure word-group similarity. In: Proceedings of the 22nd International Conference on Computational Linguistics, vol. 1. Association for Computational Linguistics (2008)
16. Zhao, Q., Rezaei, M., Chen, H., Franti, P.: Keyword clustering for automatic categorization. In: 2012 21st International Conference on Pattern Recognition (ICPR). IEEE (2012)
17. Michael Pucher, F.T.W.: Performance Evaluation of WordNet-based Semantic Relatedness Measures for Word Prediction in Conversational Speech (2004)
18. Markines, B., et al.: Evaluating similarity measures for emergent semantics of social tagging. In: Proceedings of the 18th International Conference on World Wide Web. ACM (2009)
19. Berry, M.W., Dumais, S.T., O'Brien, G.W.: Short text clustering by finding core terms. Knowledge and Information Systems 27(3), 345–365 (2011)