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].

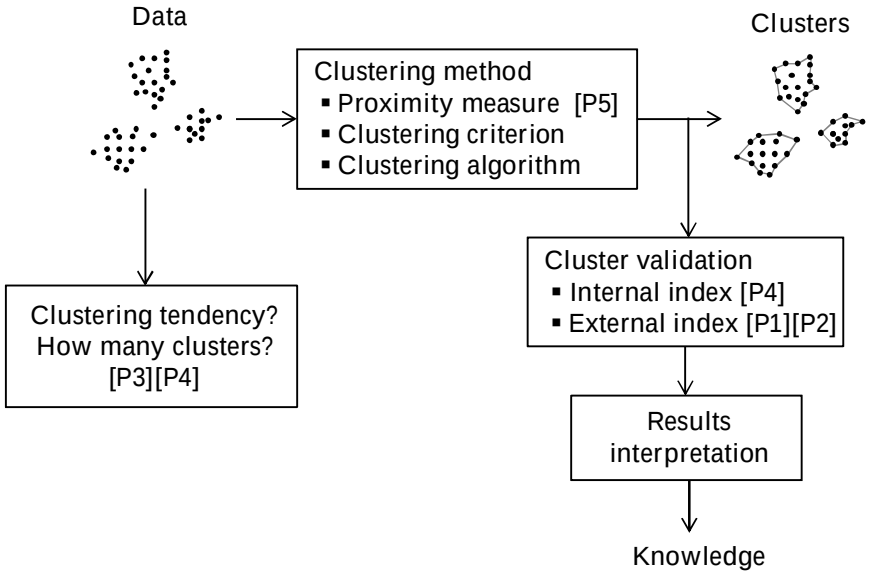


Figure 1.1: Basic components of cluster analysis

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].

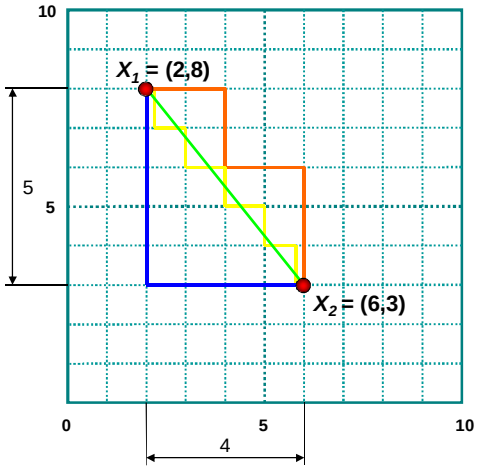
Euclidean distance is a special case (p=2) of a more general metric called Minkowski distance:

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

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.

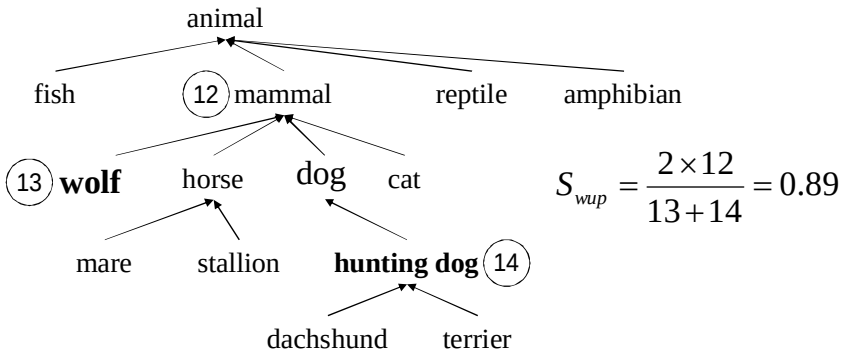


Figure 2.2: Part of WordNet taxonomy

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.

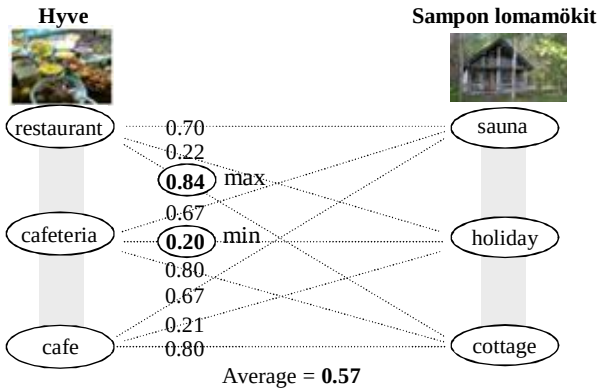


Figure 2.3: Minimum and maximum similarities between two location-based services is derived by considering two keywords with minimum and maximum similarities

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.

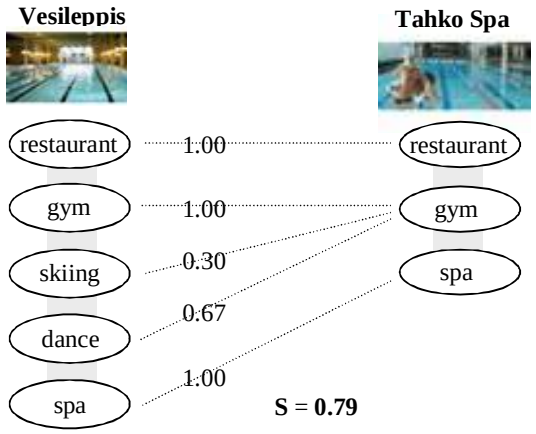


Figure 2.4: Matching between the words of two objects.

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 $O(N)$.

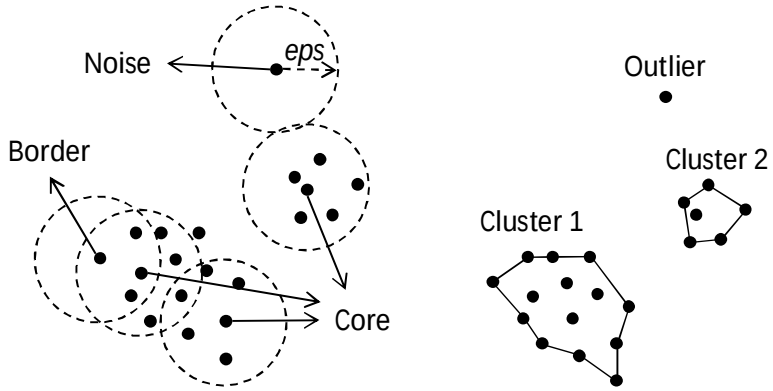


Figure 3.1: Three types of points are defined in the DBSCAN algorithm; two clusters are identified in this example, where *eps*=1 and *minPt*=5.

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).

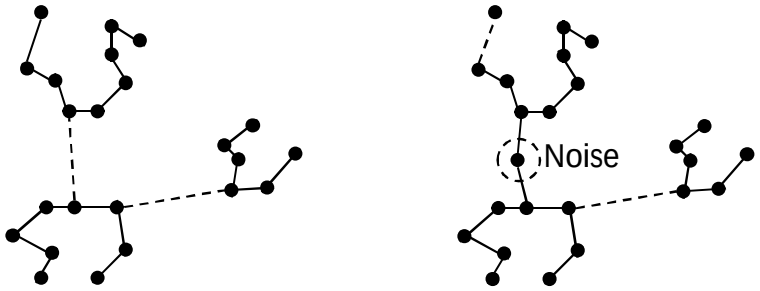


Figure 4.1: Spanning trees of clusters are used to derive the cost function.

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.

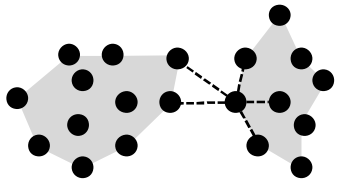


Figure 4.2: Five nearest neighbors are considered to calculate the cost function. For the selected point, two neighbors are located in the other cluster.

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.

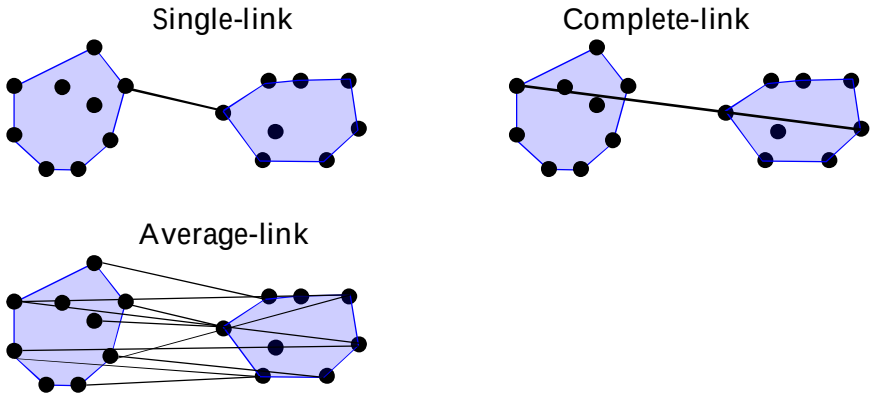


Figure 4.3: Distance between two clusters

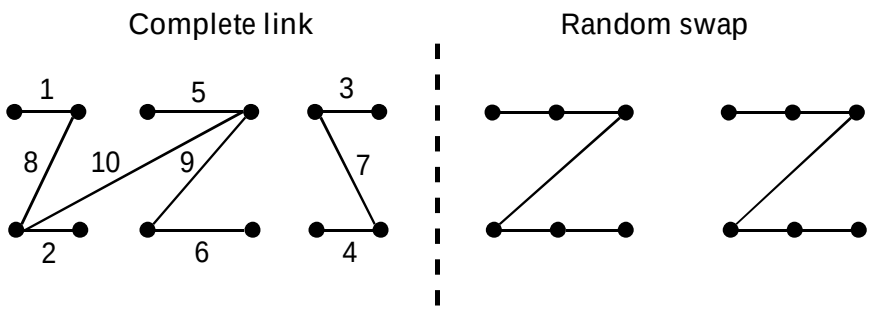


Figure 4.4: Complete link agglomerative clustering (left) results in a higher value of the cost function MAX-MAX comparing to the random swap algorithm (right). The numbers show the order of merges.

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.

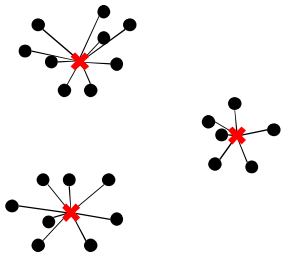


Figure 5.1: Illustration of the sum of squares within clusters

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.

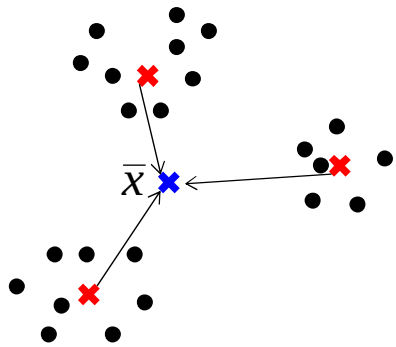


Figure 5.2: Illustration of the sum of squares between clusters.

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].

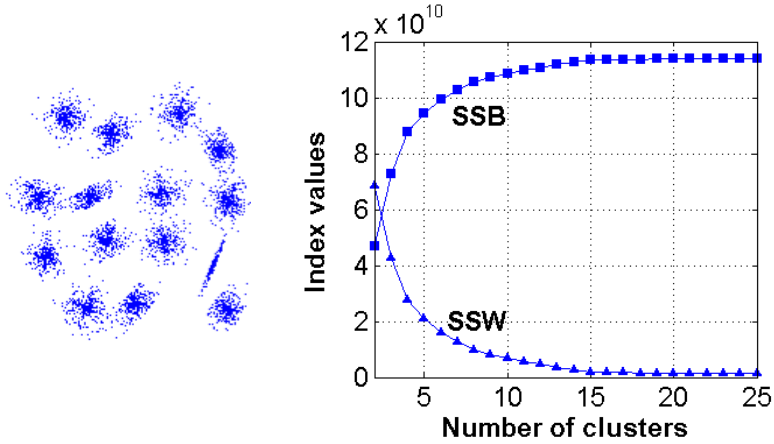


Figure 5.3: Data set S_1 (left), and the measured values of SSW and SSB (right)

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.

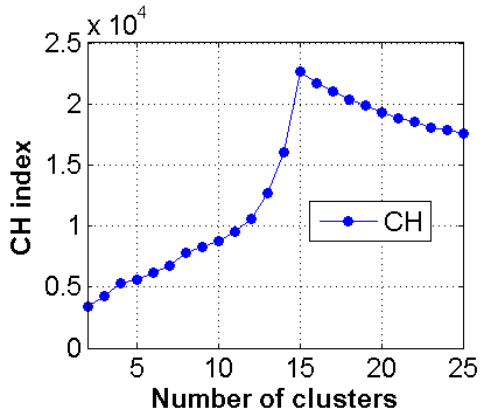


Figure 5.4: Determining the number of clusters for the data set S_1 using CH index

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].

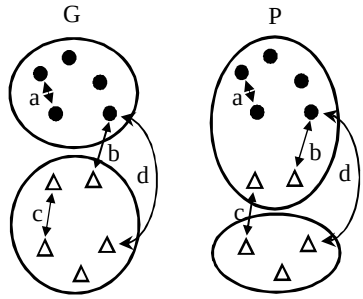


Figure 6.1: The principle of pair-counting measures.

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 $H(G|P)$:

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

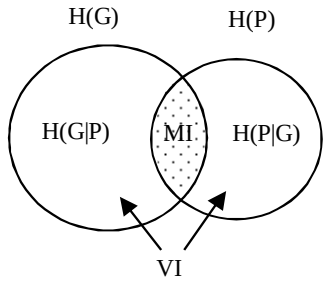


Figure 6.2: Mutual information and variation of information

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*.

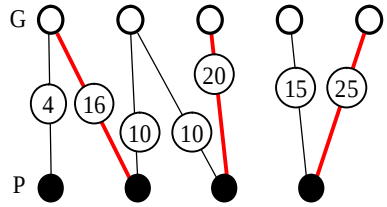


Figure 6.3: Pairing clusters to maximize overall similarity. The thick lines show the optimal pairing where the overall similarity according to number of shared objects would be (25+20+16)=61.

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).

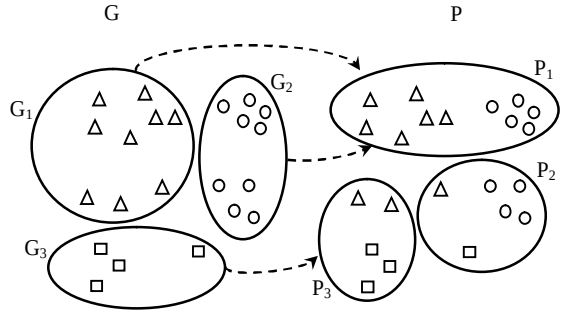


Figure 6.4: Matching clusters based on maximum shared objects. Cluster P_2 remains unmatched. In the pairing process of CH, G_2 is paired with P_2 after excluding G_1 and P_1 as the first pair.

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.

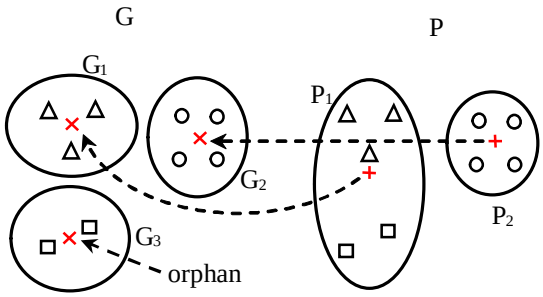


Figure 6.5: Matching centroids from *P* to *G* based on nearest neighbor mapping used in CI and CSI; one orphan centroid shows one difference in global allocation.

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.

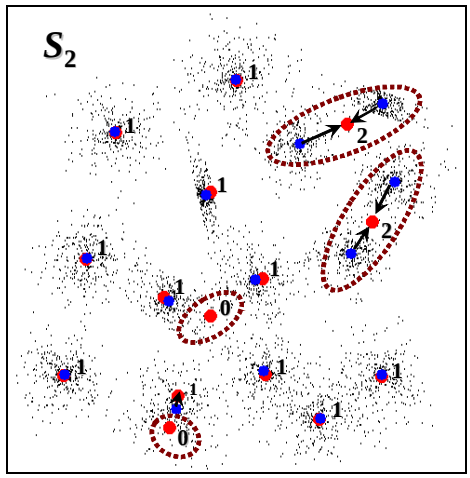


Figure 6.6: Two sets of prototypes and their mappings are shown. There are two orphans resulting in the index value of CI=2.

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.

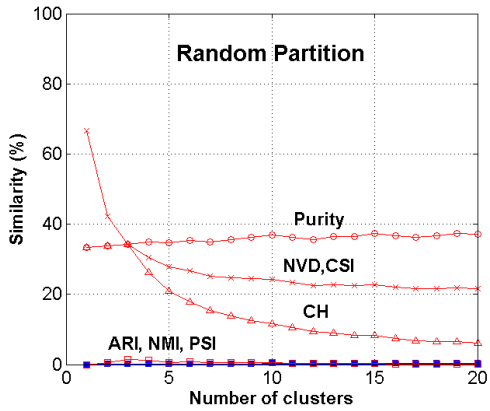


Figure 6.9: Random partitioning with different numbers of clusters in *P* from *K*=1 to 20

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.

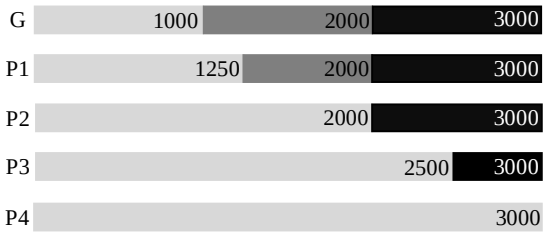


Figure 6.10: Enlarging the first (light grey) cluster in steps of 50 objects by moving the objects from the other two clusters

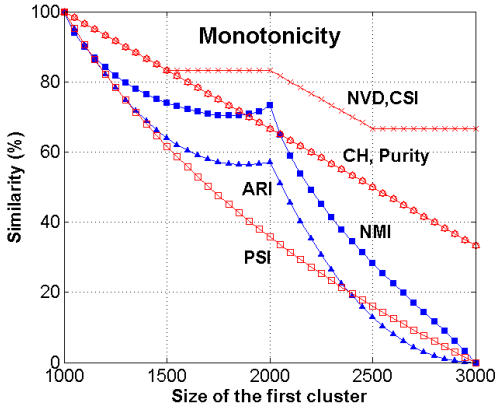


Figure 6.11: Increasing the size of the first cluster until it contains all data objects

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.

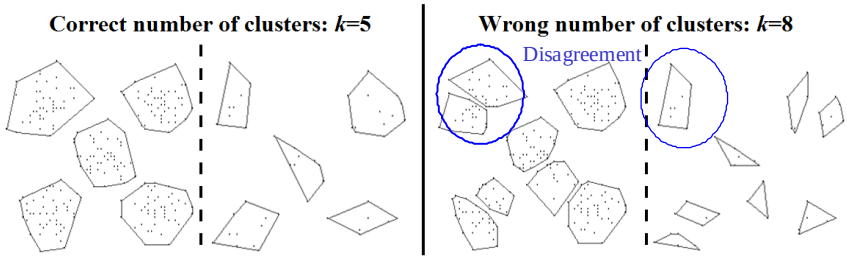


Figure 6.12: Stability-based method for finding the number of clusters. Stable (left) and unstable (right) results are produced when the correct and incorrect number of clusters are applied.

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.

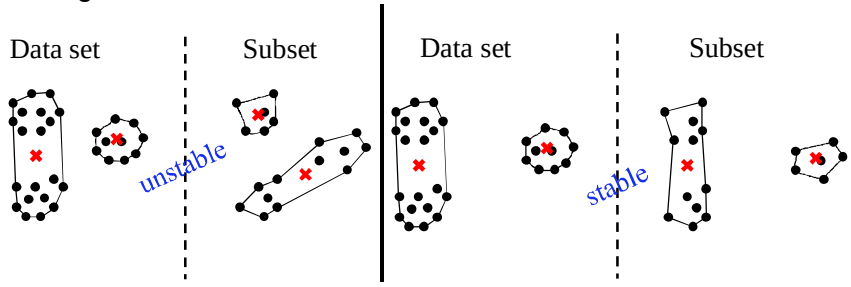


Figure 6.13: Unstable results for symmetrically and stable results for non-symmetrically positioned clusters when the incorrect number of clusters k=2 is applied.

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.

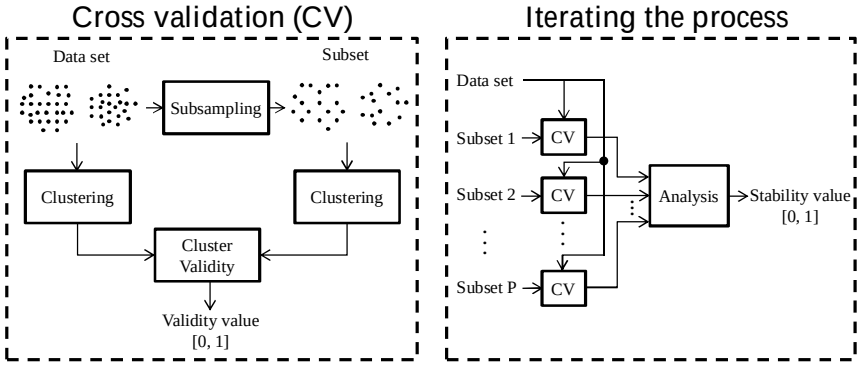


Figure 6.14: Cross-validation technique; clustering of a full data set is compared with the clustering of its subset (left). The process is repeated for a number of subsets (right).

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.

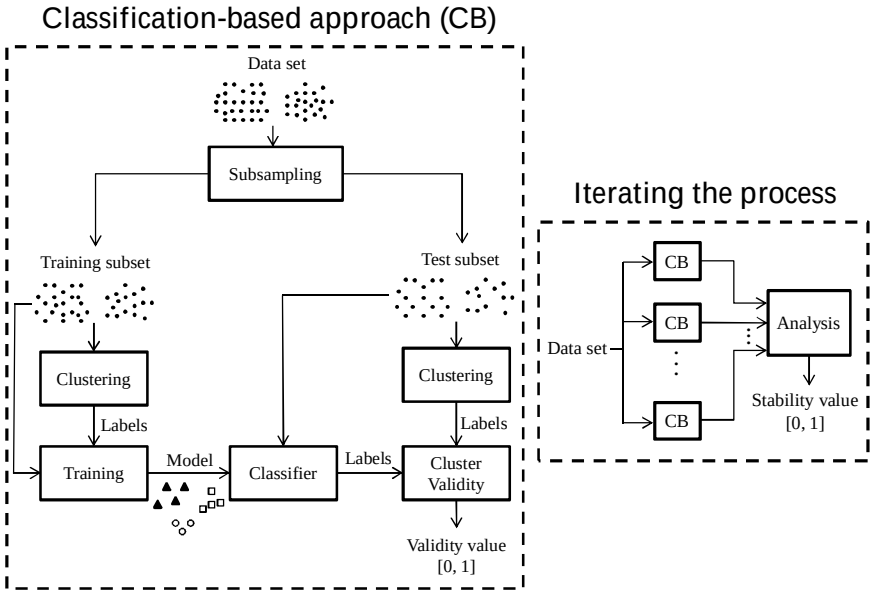


Figure 6.15: Classification-based approach (left), and iterating the process for several train and test sets (right).

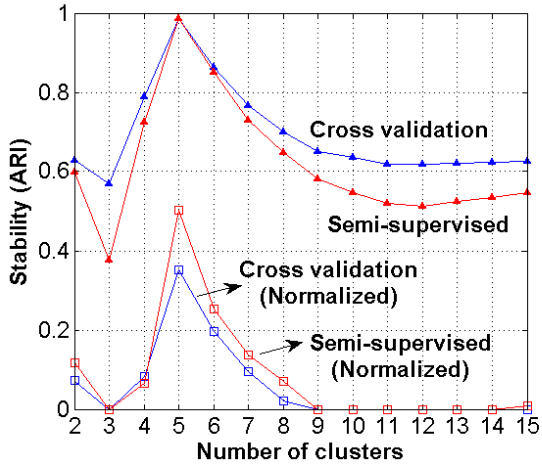


Figure 6.16. Example of stability-based method for the data set in Figure 6.12. 100 subsets are used in the cross-validation approach, each 20% of the full data set. The sizes of train and test sets in the classification-based approach are 80% and 20%. Random swap algorithm is used for clustering [25]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