

External Validation Measures for Nested Clustering of Text Documents

Karol Draszawka, Julian Szymański

Department of Computer Systems Architecture,
Gdańsk University of Technology, Poland,
kadr@eti.pg.gda.pl
julian.szymanski@eti.pg.gda.pl

Abstract. This article handles the problem of validating the results of nested (as opposed to "flat") clusterings. It shows that standard external validation indices used for partitioning clustering validation, like *Rand statistics*, *Hubert Γ statistic* or *F-measure* are not applicable in nested clustering cases. Additionally to the work, where *F-measure* was adopted to hierarchical classification as *hF-measure*, here some methods to get desired *hRand* and *h Γ* indices for nested clustering are presented. Introduced measures are evaluated and, as an exemplary application, a validation of nested clustering methods for Wikipedia articles using OPTICS algorithm is shown. Clustering validation, Rand index, Hubert's index, F-measure, OPTICS, reachability plot.

1 Introduction

One of the fundamental *data mining* tasks is clustering [1]. The goal of a clustering algorithm is to find similar objects in a dataset and group them into clusters. Such an algorithm tries to minimize intra-cluster distance while maximizing inter-cluster distance. In the context of text documents, such clustering is very important, because finding interesting information within still growing large text repositories is a rather difficult task, if the repository is not properly organized. Clustered (categorized) collection of texts is much easier to work with. It would be especially appreciated, if clusters that still contain many articles are also organized in sub-clusters, so that the whole repository has a structure of a folder-tree. This kind of organizing data into a tree of nested clusters can be achieved using some methods of *hierarchical clustering* [2], which produce *dendrograms*¹ and then pruning them, or other algorithms that create folder-like nested trees directly. We group all those methods under the general *nested clustering* label.

Having got a partition (in our case it is a hierarchical partition) of data elements into clusters, we must be sure that this organization of objects is

¹ A dendrogram is a special kind of cluster tree with all N objects of a dataset as N singleton leaf clusters and $N - 1$ non-leaf clusters, which are unions of two clusters from lower levels.

valid, which means that it really groups similar and separates dissimilar objects. Particularly, the organization of a text repository is valid, when clusters (i.e. categories of articles) contain texts with the same or correlated subjects. This *validation* step is fundamental for achieving reliable clustering results and has become a topic of separate research examination. There are three main approaches to cluster validation. They are based on internal, relative and external criteria [3],[4],[5].

Internal validation techniques employ the fact that clusters are sets of objects that are compact and well separated. They measure compactness – distances between objects in the same clusters (if they are smaller, clusters are more compact) and separation – distances between clusters (bigger distances indicates better separation). This idea can be applied in many ways, thus different internal validation indices have been proposed. The classic paper [6] presents the examination of 30 internal validation indices. More recent papers [3],[4],[7],[8],[9] continue to find best indices among already known and new ones. Unfortunately, the conclusion often repeated from these studies is that there is no best internal validation index, because they are all data dependent.

Relative approach to cluster validation relies on repeating the same clustering algorithm multiple times using different parameters, and choosing the most stable results. For example, if the number of clusters is one of the input parameters, one tries clusterings using various number of clusters and chooses that for which internal indices are best. If the number of clusters is not the parameter, relative validation allows one to choose the parameter values that are in the middle of the broadest range for which the number of clusters is constant [4].

External validation may be used when the real partition of the clustered data is known a priori. Knowing the classes (or categories) of the data objects, we can compare them with the clusters created by an algorithm. It is known [3], that external validation is more accurate than internal or relative. This is the type of validation especially important, when one tries to find the best clustering method for a specific task and usually uses a variety of algorithms on a certain dataset with good known class structure.

This article is devoted to external validation measures for nested clustering and is organized as follows. Section 2 describes popular external validation indices for non-nested 'flat' clustering. Then, section 3 introduces modified versions of these indices, so they can properly evaluate nested clustering quality. That section also discusses what properties such measures should have. Some of the measures presented there comes from the literature, some are invented by us. In section 4 we present an evaluation of these measures based on experiments on artificial data sets. The best measures are then presented in a practical context in section 5, in which we are comparing different cluster extraction methods from reachability plots for automatic Wikipedia articles nested clustering. Section 6 concludes the presented material.

2 External Validation Indices for "Flat" Clustering

In this section we present three popular external validation measures, namely *F-measure*, *Rand statistic* and *Hubert Γ statistic*, in their normal form adequate only for non-nested "flat" clustering tasks. In the next section, we show their modified versions suited to cluster hierarchies. The first one index – *F-measure* – is a well known tool in the information retrieval domain, but it is also used as a measure of partitioning quality, where it is known to be better than such factors as *purity*, *coverage* or *entropy* [10]. *Rand statistic* and *Hubert Γ statistic*, originally introduced in the classical papers [11, 12], are recommended in [3] [13], where the authors show that the validation using these factors is more accurate than those with *Jaccard coefficient* or *Fowlkes and Mallows index*.

For external validation we must assume that, for a given set of objects X , we have both: the real, true partition of this set $C^T = \{C_1^T, \dots, C_{K^T}^T\}$ (we can call C_k^T sets as *classes*, K^T is the number of classes) and clustering partition, the result of clustering or classification algorithm, $C^C = \{C_1^C, \dots, C_{K^C}^C\}$ (C_k^C sets are *clusters*, K^C is the number of clusters). Having these two partitions we can compute how similar they are.

2.1 F-measure

F-measure is a mixture of two indices: *precision* (P), which measures the homogeneity of clusters with respect to a priori known classes, and *recall* (R), that evaluates the completeness of clusters relatively to classes. Having the previously introduced notation, *precision* of cluster C_k^C with regard to class C_l^T is computed as follows:

$$P(C_k^C, C_l^T) = \frac{\#(C_k^C \cap C_l^T)}{\#C_k^C}. \quad (1)$$

Recall of cluster C_k^C with respect to class C_l^T is defined as:

$$R(C_k^C, C_l^T) = \frac{\#(C_k^C \cap C_l^T)}{\#C_l^T}. \quad (2)$$

Then, F value of the cluster C_k^C with respect to class C_l^T is, in general, the combination of these two:

$$F(C_k^C, C_l^T) = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 \cdot P + R}, \quad \beta \in [0, +\infty), \quad (3)$$

but most often researchers treat precision and recall with the same weights putting $\beta = 1$ and then F value is the harmonic mean of the *precision* and the *recall*:

$$F(C_k^C, C_l^T) = \frac{2P(C_k^C, C_l^T)R(C_k^C, C_l^T)}{P(C_k^C, C_l^T) + R(C_k^C, C_l^T)} = \frac{2}{\frac{1}{P(C_k^C, C_l^T)} + \frac{1}{R(C_k^C, C_l^T)}}. \quad (4)$$

The F-measure for cluster C_k^C is the highest of F values obtained by comparing this cluster with each of known classes:

$$F(C_k^C) = \max_{C_l^T \in C^T} F(C_k^C, C_l^T). \quad (5)$$

Finally, the *F-measure* of the whole clustering is a weighted sum of individual F-measures of all clusters:

$$F(C^C) = \sum_{C_k^C \in C^C} \frac{\#C_k^C}{N} F(C_k^C). \quad (6)$$

where N is the number of all objects in the dataset.

2.2 Measures Based on Similarity Matrices

For both partitions, C^T and C^C , we can calculate a binary $N \times N$ *similarity matrix* \mathbf{S} :

$$\mathbf{S} = [s_{i,j}], \quad s_{i,j} = \begin{cases} 1 & \text{if } x_i \text{ and } x_j \text{ are in the same class/cluster,} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Hubert Γ statistic. *Hubert Γ statistic* measures the correlation of the partitions C^T and C^C based on the correlation between their respective similarity matrices \mathbf{S}^T and \mathbf{S}^C :

$$\Gamma = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^N s_{i,j}^T \cdot s_{i,j}^C, \quad (8)$$

where $M = n(n-1)/2$ is the total number of pairs of different data objects in the dataset. However, it is better to use the normalized version of *Hubert Γ statistic*, which is a Pearson product-moment correlation coefficient (PMCC):

$$\Gamma^* = \text{PMCC}(\mathbf{S}^T, \mathbf{S}^C) = \frac{1}{(M-1)\sigma^T\sigma^C} \sum_{i=1}^{N-1} \sum_{j=i+1}^N (s_{i,j}^T - \mu^T)(s_{i,j}^C - \mu^C), \quad (9)$$

where μ^T , μ^C , σ^T and σ^C are the respective sample means and standard deviations of the values in \mathbf{S}^T and \mathbf{S}^C .

Rand statistic. *Rand statistic* employs the fact that for each pair of data objects x_1 and x_2 ($x_1 \neq x_2$) from the data set, we have one of four possible situations:

- (a) x_1 and x_2 are in the same class/cluster in both C^T and C^C ,

- (b) x_1 and x_2 are from the same class (in C^T) but have been clustered to different clusters in C^C ,
- (c) x_1 and x_2 are from different classes (in C^T) but have fallen to the same cluster in C^C ,
- (d) x_1 and x_2 are in different classes/clusters in both C^T and C^C .

The number of situations (a), (b), (c) and (d) indicates factor values a , b , c and d respectively. More similar C^T and C^C are, bigger are a and d factors. This observation leads to the definition of *Rand statistic* calculated as follows:

$$\text{Rand} = \frac{a + d}{M}, \quad (10)$$

where M denotes the same as in previous equations (8) and (9).

It may be noted, that a and d factors can be easily calculated from similarity matrices \mathbf{S}^T and \mathbf{S}^C :

$$a = \# (s_{i,j}^T = 1 \wedge s_{i,j}^C = 1)_{i \in [1,N], j > i}, \quad (11)$$

$$d = \# (s_{i,j}^T = 0 \wedge s_{i,j}^C = 0)_{i \in [1,N], j > i}. \quad (12)$$

Therefore, the nominator in the equation 10 can be expressed in terms of the similarity matrices and *Rand* index can be calculated as:

$$\text{Rand} = \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N (1 - |s_{i,j}^T - s_{i,j}^C|)}{M} = 1 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^N |s_{i,j}^T - s_{i,j}^C|}{M}. \quad (13)$$

Rand statistic and *F-measure* take values between 0 and 1, *Hubert Γ^* statistic* – between -1 and 1. For all three measures, higher value indicates better clustering with value 1 for perfect adequacy between known classes and clusters created by an algorithm.

3 External Validation Indices for Nested Clustering

3.1 Requirements of Nested/Hierarchical Validation Measure

For nested clustering, an external validation index should have the ability to discriminate small misclassifications, i.e. a situation when an object is put into a wrong class which is however not far in the hierarchy from the right one, from bigger misclassifications. Kiritchenko et al. [14] have formulated three requirements that a hierarchical evaluation measure should satisfy:

1. The measure M gives credit to partially correct classification, e.g. misclassification into node 4 when the correct node is 6 (figure 1) should be penalized less than misclassification into node 2. We can write this as: $M(C_{6 \rightarrow 4}) > M(C_{6 \rightarrow 2})$.
2. The measure punishes distant errors more heavily: e.g. $M(C_{6 \rightarrow 1}) < M(C_{6 \rightarrow 3})$ and $M(C_{6 \rightarrow 4}) < M(C_{6 \rightarrow 1})$.

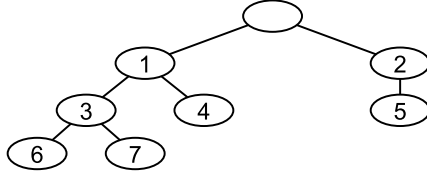


Fig. 1: An exemplary class/cluster hierarchy.

3. The measure gives more penalty points to misclassifications at higher levels of hierarchy (closer to the root) than that at lower levels, e.g. $M(C_{1 \rightarrow 2}) < M(C_{3 \rightarrow 4}) < M(C_{6 \rightarrow 7})$.

Although these requirements concern validation of hierarchical *classification* task, they should also be satisfied by external validation measures of hierarchical *clustering*. But in this second case, because of the fact that nested clusters returned by an algorithm can be of different number and hierarchical structure than the true classes are, we should formulate additional requirements:

4. If a class is split into some number of clusters, then the punishment for that is smaller when the number of superfluous clusters are small and they are closely connected in the hierarchy.
5. Analogously, if some number of classes are joined into one cluster, the score of the matching should be dependent on the relations between joined classes in the true class hierarchy.

3.2 hF-measure

To meet the requirements they listed (req. 1-3), Kiritchenko et al. [14] have adopted *F-measure* to nested classification by calculating *precision* and *recall* (formulas (1) and (2)) with respect to the rule that objects are in a given class/cluster if they are exactly in that class/cluster *or* in some of ancestors, except the root (because, in this way, every object belongs to the root).

These enhanced precision and recall values can be efficiently computed using *classification arrays* [15, 16]. A classification array have the form of a binary matrix \mathbf{A} , where each row corresponds to a data object, each column represents a different cluster (the size of the matrix is $N \times K$) and element $a_{i,j}$ takes 1 when there is an assignment of object i to cluster j , and 0 otherwise. A fragment of the classification array for the graph in figure 2 is presented in table 1.

Having the true classification array \mathbf{A}^T and returned from clustering \mathbf{A}^C , formulas for *precision* and *recall* can be expressed as:

$$P(C_k^C, C_l^T) = \frac{\mathbf{a}_{:,k}^C \cdot \mathbf{a}_{:,l}^T}{\sum_{i=1}^N \mathbf{a}_{i,k}^C}, \quad (14) \quad R(C_k^C, C_l^T) = \frac{\mathbf{a}_{:,k}^C \cdot \mathbf{a}_{:,l}^T}{\sum_{k=i}^N \mathbf{a}_{i,l}^T}. \quad (15)$$

3.3 Cophenetic and Cladistic Coefficients

The two classic measures that can be used for external validation of nested clustering are *cophenetic* and *cladistic* coefficients [15, 16]. Originally, they were invented to measure the quality of dendrograms from full hierarchical clusterings, but they also can be applied to folder-like tree structures when adopted in a way presented below. They evaluate the correlation between *dissimilarity matrices* \mathbf{D}^T and \mathbf{D}^C , whose elements $d_{i,j}$ indicate a kind of distance between positions of objects i and j in the true tree of classes (in the case of \mathbf{D}^T) and between their positions in the tree of clusters returned from a nested clustering (\mathbf{D}^C).

Cophenetic distance between objects in a general class/cluster tree (not only dendrograms) can be formulated as the height of a subtree which joins those objects:

$$d_{coph}(i, j) = \max(\text{lev}(i, \text{NCC}), \text{lev}(j, \text{NCC})) , \quad (16)$$

where *NCC* is the abbreviation of the *Nearest Common Cluster* and *lev* function denotes the level-based distance between clusters in which objects i and j are.

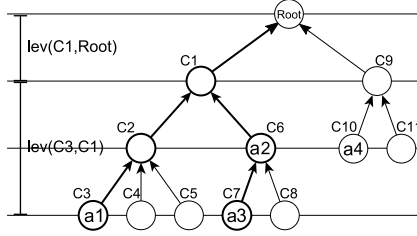


Fig. 2: An exemplary nested clustering with not correctly assigned objects $a1 - a4$ which came from the same class.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11
a1	1	1	1	0	0	0	0	0	0	0	0
a2	1	0	0	0	0	1	0	0	0	0	0
a3	1	0	0	0	0	1	1	0	0	0	0
a4	0	0	0	0	0	0	0	0	1	1	0

Table 1: Classification array (\mathbf{A}^C) for the example given in figure 2.

For example, in figure 2 objects $a1$ and $a2$ are joined together in cluster $C1$, and the subtree containing nodes $C1$, $C2$, $C3$ and $C6$ is of height 2, so the cophenetic distance between objects $a1$ and $a2$ is 2. Similarly we have $d_{coph}(a1, a3) = 2$ and $d_{coph}(a1, a4) = 3$.

Cladistic distance between two objects in a class/cluster tree is the number of line segments that separate clusters which those objects are:

$$d_{clad}(i, j) = \text{lev}(i, \text{NCC}) + \text{lev}(j, \text{NCC}) . \quad (17)$$

Using the example from figure 2, $d_{clad}(a1, a2) = 3$, $d_{clad}(a1, a3) = 4$ and $d_{clad}(a1, a4) = 5$.

The external validation measures of nested clusterings are then product moment correlation coefficients, *Coph* or *Clad*, between dissimilarity matrices \mathbf{D}^T and \mathbf{D}^C :

$$Coph = \text{PMCC}(\mathbf{D}_{coph}^T, \mathbf{D}_{coph}^C) , \quad (18) \quad Clad = \text{PMCC}(\mathbf{D}_{clad}^T, \mathbf{D}_{clad}^C) . \quad (19)$$

3.4 Measures Based on Non-binary Similarity Matrices

In subsection 2.2 we present a similarity matrix \mathbf{S} (eq. (7)) which takes only binary values. But if a validation of non-perfect nested clustering should be able to distinguish small misassignments from the bigger ones, then we need to operate with a non-binary similarity matrix \mathbf{S}^* .

There can be many ways of defining such a matrix. First, we can derive it from cophenetic or cladistic dissimilarity matrix (eq. (16) and (17)), using the following formulas:

$$s_{coph}^*(i, j) = 1 - \frac{d_{coph}(i, j)}{treeHeight}, \quad (20) \quad s_{clad}^*(i, j) = 1 - \frac{d_{clad}(i, j)}{2 \cdot treeHeight}, \quad (21)$$

where *treeHeight* is the height (in levels) of the whole nested class/cluster structure. Resulting similarity matrix takes values between $[0, 1]$ with 1 for objects from the same class/cluster and values smaller than 1 for objects from different clusters – proportionally to the distance between these clusters in the tree.

We provide another way of calculation a non-binary similarity matrix for nested clustering, using formula:

$$s_{ca}^*(i, j) = \frac{\text{lev}(\text{NCC}, \text{Root})}{\text{mean}(\text{lev}(i, \text{Root}), \text{lev}(j, \text{Root}))}, \quad (22)$$

We named this method *CA*, because it is the ratio between the number of *common* clusters of two objects to the number of *all* the clusters into which objects fell (because they can be in different number of clusters, the arithmetic mean is taken). If $s_{ca}^*(i, j)$ is undefined (when both i and j objects are not clustered and they belong only to the root), then we propose to put $s_{ca}^*(i, j) = 0$.

For our example presented in figure 2, the similarity between objects $a1$ and $a2$, assigned to clusters C3 and C6 respectively, is calculated according to their nearest common cluster (C1). The level distance between NCC and the root is $\text{lev}(C1, \text{Root}) = 1$, the distances from C3 and C6 to the root are $\text{lev}(C3, \text{Root}) = 3$, $\text{lev}(C6, \text{Root}) = 2$. Using formula (22), $s_{ca}^*(a1, a2) = \frac{1}{\text{mean}(2, 3)} = 0.4$. By contrast, $s_{ca}^*(a1, a3) = \frac{1}{3}$, and for objects $a1$ and $a4$, the nearest common cluster is the root, so $s_{ca}^*(a1, a4) = 0$.

CA similarity matrix can be effectively calculated using classification arrays. We can rewrite expression (22) as (23):

$$s_{ca}^*(i, j) = \frac{2}{2} \cdot \frac{\text{lev}(\text{NCC}, \text{Root})}{\frac{\text{lev}(i, \text{Root}) + \text{lev}(j, \text{Root})}{2}} = 2 \cdot \frac{\mathbf{a}_{i,\cdot} \cdot \mathbf{a}_{j,\cdot}}{\sum_{k=1}^K a_{i,k} + \sum_{k=1}^K a_{j,k}}. \quad (23)$$

A hierarchical version of Rand validation index can be then easily obtained by putting into expression (13) the augmented non-binary similarity matrices \mathbf{S}^{*T} and \mathbf{S}^{*C} instead of traditional binary ones. We can name this hierarchical version of Rand index as *hRand* analogously to *hF-measure* proposition by Kiritchenko et al. Because similarity matrices employed in *hRand* calculation can be of three types (eq. (20), (21) and (23)), in fact we have not one, but three *hRand* validation indices: $hRand_{coph}$, $hRand_{clad}$, $hRand_{ca}$.

In the same way, we have three types of PMCC-based hierarchical normalized Hubert’s hI_{coph}^* , hI_{clad}^* , hI_{ca}^* coefficients by replacing traditional binary similarity matrices in formula (9) with one of the three non-binary ones. However, because PMCC measures the linear dependence between variables, it has the same value for similarity or dissimilarity matrices – the transformations from dissimilarity to similarity matrix (eq. (20) and (21)) are linear. Therefore, hI_{coph}^* , calculated from similarity matrices, and $Coph$ coefficient, obtained from dissimilarity matrices, are the same. Analogously $hI_{clad}^* = Clad$.

All the measures discussed in the previous and this section are presented in table 2.

Table 2: External validation indices for ‘flat’ and nested clustering.

		(dis-)similarity matrix based	
		Rand($\mathbf{S}^T, \mathbf{S}^C$)	PMCC($\mathbf{S}^T, \mathbf{S}^C$)
‘flat’ clustering	F-measure	Rand	Hubert’s I^*
nested clustering	hF-measure	$hRand_{coph}$	$I_{coph}^* = Coph$
		$hRand_{clad}$	$I_{clad}^* = Clad$
		$hRand_{ca}$	I_{ca}^*

4 Evaluation of the Measures Using Synthetic Data

In this section we present experiments which were performed to see how the introduced external validation indices for nested clustering work: whether they satisfy the conditions of a hierarchical measure (see subsection 3.1) and, if so, how they are sensitive to classification errors at different levels of a hierarchy.

4.1 The Same Class and Cluster Trees

First experiments were done to see whether the indices fulfil 3 basic properties of a hierarchical *classification* measure. We prepared an exemplary true nested classification C^T , presented in figure 3. A binary tree of level 3 was chosen, because many cases of misclassification can be shown on such a generic structure. Every node of that tree is a class that has 5 objects directly associated with it (non-leaf classes has also objects indirectly associated with them through child classes).

Then, a series of non-perfect classification trees C^C was artificially prepared, by taking all 5 objects from class 15 and assigning them to one of the other classes X ($X \in 1, 2, \dots, 14$). Such a classification is denoted by $C_{15 \rightarrow X}$. It was done to simulate different misclassification cases – from not very huge, like $C_{15 \rightarrow 14}$, to rather significant, for example $C_{15 \rightarrow 2}$. All indices were computed to measure the similarity between C^T and $C_{15 \rightarrow X}$.

If a measure M fulfils the Kiritchenko et al. requirements, then the following inequalities should be true:

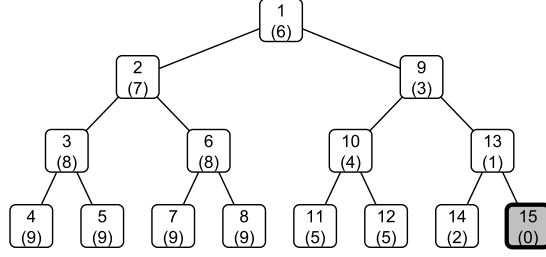


Fig. 3: The structure of C^T . Numbers inside circles describe labels of classes. Numbers in parenthesis indicated the expected ranking of nodes wrt. the importance of misclassification from C_{15} : less value, less crucial misclassification.

1. $M(C_{15 \rightarrow \{9-14\}}) > M(C_{15 \rightarrow \{2-8\}})$
2. $M(C_{15 \rightarrow 13}) > M(C_{15 \rightarrow 9}) > M(C_{15 \rightarrow 1})$
 $M(C_{15 \rightarrow 13}) > M(C_{15 \rightarrow 14})$
 $M(C_{15 \rightarrow 9}) > M(C_{15 \rightarrow \{10-12\}})$
 $M(C_{15 \rightarrow 1}) > M(C_{15 \rightarrow \{2-8\}})$
3. $M(C_{15 \rightarrow 14}) > M(C_{13 \rightarrow 10}) > M(C_{9 \rightarrow 2})$

Conditions 1 and 2 mean that the validation measure should decrease following the order presented by numbers in parenthesis in figure 3.

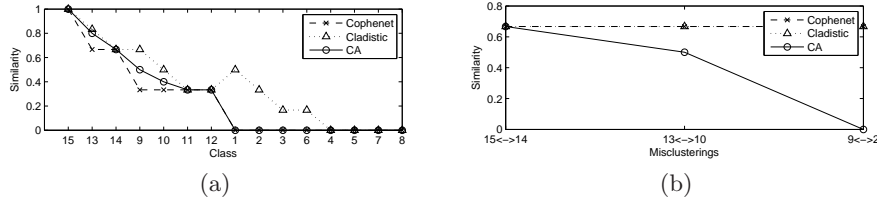


Fig. 4: (a) The values of similarity matrix between object from class 15 and objects from other classes. (b) The values of similarity matrix between objects from indicated classes.

We have first checked, if similarity matrix elements have expected values. Figure 4a shows that similarity matrix based on *cophenetic* values and on *CA* method are non-increasing, which is desirable when ordered as presented. However *cophenetic*-type line is more stair-like – this kind of similarity equalize $C_{15 \rightarrow 13}$ with $C_{15 \rightarrow 4}$ and $C_{15 \rightarrow 9}$ with $C_{15 \rightarrow \{10, 11, 12\}}$. *CA*-similarity do not have this drawback. *Cladistic*-type similarity equals $C_{15 \rightarrow 1}$ with $C_{15 \rightarrow 10}$, which is not correct, because the second classification has a partially true assignment. What is even worse is that it penalizes $C_{15 \rightarrow 1}$ less than $C_{15 \rightarrow 11}$ or $C_{15 \rightarrow 12}$. An advantage of this type of similarity matrix is that it distinguishes between bad and

even worse misclassifications, when the other two give 0 for all binds to wrong main branch of the class tree.

Figure 4b depicts that only *CA*-type similarity matrix fulfils the third requirement of hierarchical measure, that a penalty for misclassification depends on the level of a hierarchy at which this mistake was done².

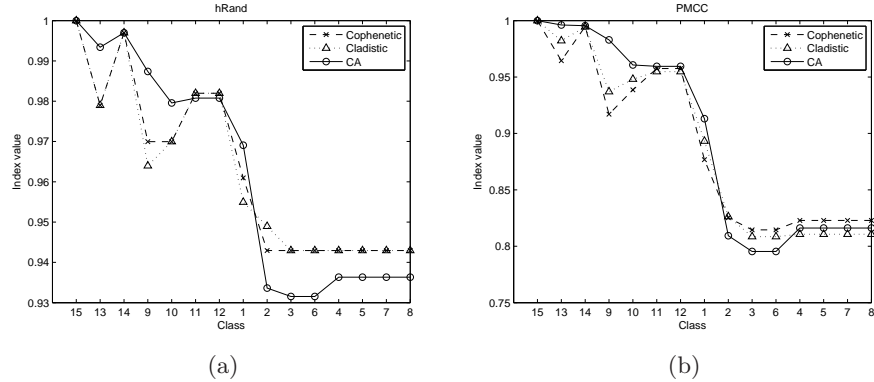


Fig. 5: hRand (a) and PMCC (b) indices for non-perfect clusterings $C_{15 \rightarrow X}$.

The charts in figure 5 show how similarity-based indices perform depending on the type of similarity matrices. For *CA*-based indices, especially $PMCC_{CA}$, graphs are almost always decreasing, which means that they correctly evaluate the significance of misclassification for partially correct cases (into classes 9 to 14). *Cophenet* and *cladistic*-based indices tend to give higher scores when objects are deeply classified, which violates the second Kiritchenko et al. requirement. All measures punish severely far misclassifications (into other main branch of a hierarchy), but not distinguish between them correctly. *PMCC* more dynamically changes values than *hRand* (because its range is two times greater than the range of *hRand*). Figure 6 illustrates that despite the fact that *cophenetic* and *cladistic* similarities do not distinguish levels of mistakes, the *hRand* and *PMCC* indices based on them are able to do that. This is due to the fact that the relationships between all the objects are taken into account when an index is calculated, and this results in slightly smaller indices values for more significant misclassifications. However, *CA*-based indices more markedly express these differences.

Figure 7 shows how the *hF-measure* performs. What is not desired here is that the misclassifications $C_{15 \rightarrow 1}$ and $C_{15 \rightarrow 2}$ have smaller punishment than

² It must be emphasised that original cophenetic coefficient for *dendrograms* is able to distinguish the level of a misclassification [15]. Results presented here stem from our usage of cophenetic distance to folder-like nested clusters as a level of a subtree that join two clusters (eq. (16)).

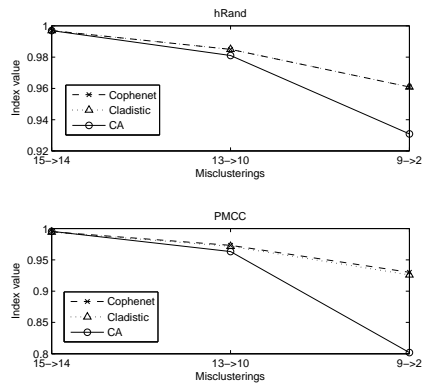


Fig. 6: Third requirement test results.

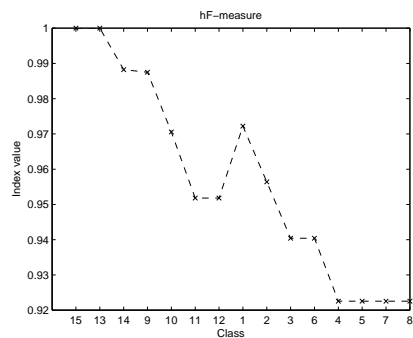


Fig. 7: hF-measure performance.

$C_{15 \rightarrow 11,12}$. For *hF-measure* it is better to misclassify into totally wrong branch of the tree but not far from the root class, than to misclassify into class far away from the root even if it is in the correct main branch. This is because mistaken objects reduce the precision not only of the class to which it directly belongs, but also of all ancestor classes. Also, *hF-measure* do not discriminate between correct classification and the situation when a whole class is classified into its parent – that is why $F(C_{15 \rightarrow 13}) = 1$.

4.2 Split and Joined Classes

The aim of the second group of experiments was to check whether presented measures satisfy two additional requirements for external validation of nested clusterings, that is how it should evaluate clusterings with not exactly the same overall hierarchy of clusters as the hierarchy of classes. This often happens, especially in situations when objects of the same class are split into a number of clusters or objects from different classes are joined into only one cluster.

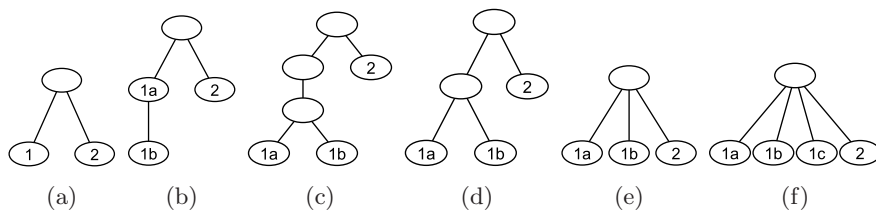


Fig. 8: The class tree (a) and 5 cluster trees (b-f) of different non-perfect clusterings ordered in increasing misclustering error significance.

Tests were made using 24 objects of 2 classes, 12 objects per class (fig. 8a) and then creating 5 types of wrong clusterings (fig. 8b-8f). Each of them mistakenly splits objects of class 1 equally into two or three clusters. Depending on the position of the new clusters in a clustering tree, measures should give bigger or smaller matching score. Intuitively, the score should be decreasing for clusterings ordered as presented in figure 8.

Using the same data, we have also checked how measures works when two or three classes are joined into one cluster. Clusterings (fig. 8b-8f) were treated as true class hierarchies and a tree 8a served as a partially incorrect clustering. Actually, for validation measures based on similarity matrices it is indifferent which of the assignments is the reference and which is the one under evaluation. Therefore, only the hF-measure reacts differently in situations with joining and splitting errors.

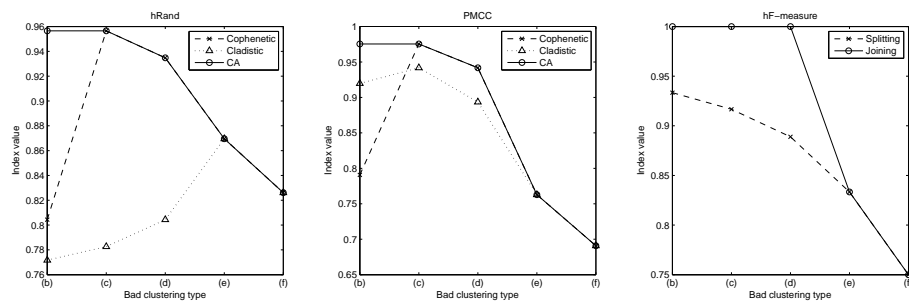


Fig. 9: hRand, PMCC and hF-measure for splitting and joining classes errors for structures from fig. 8.

Figure 9 shows the results. We can see that both hRand and PMCC validation indices work correctly only with CA-type of similarity matrices. Cophenetic and Cladistic-based measures underrate 8b situation. In addition, $hRand_{clad}$ performs especially improperly. The hF-measure judges as expected only splitting kind of errors. First three of joining misclusterings are not found by hF-measure, because of the property, mentioned earlier, that it does not recognize situations when a whole class is classified into its parent.

5 Experiments with Text Documents Clustering

In this section, we present our usage of $hRand_{ca}$, hI_{ca}^* and $hF-measure$ – the most promising of discussed validation measure – in a practical field, which is text documents clustering.

5.1 Data Characteristics




We research methods for effective categorizing of textual content. The goal is to organize text documents sets into meaningful category trees. To evaluate our approach we find Wikipedia a very useful source of data. It provides articles that are to be organized, as well as offering categories that can be used as reference classes to perform external validation.

We obtained from Polish Wikipedia articles that fell into selected categories, and using them formed 12 datasets that are described in table 3. We selected three types of datasets ordered by ascending categorization complexity:

- **A** - datasets that contain 4 categories not related to each other,
- **B** - datasets containing parent category and 3 subcategories,
- **C** - datasets containing hierarchy of 4 categories.

Articles were represented using standard Vector Space Model (VSM) with TF/IDF (Term Frequency / Inverse Document Frequency) [17] term weighting method and cosine metric [1]. We cannot expect that the results of clustering algorithms on such pure representations of articles will be exactly the same as Wikipedia categories without the indication of the most important features. Therefore we performed a supervised feature selection based on Fisher Linear Discriminant Analysis (FLDA) [18]. Because of high data dimensionality, we also performed Principal Component Analysis (PCA) [19] to identify the most significant features in the data.

Table 3: Wikipedia categories used in experiments.

category structure	1	2	3	4
A 	Physics Arts Political science Computer science	Geography Musicology Journalism Biotechnology	Chemistry Archival science Psychology Optics	Astronomy Demography Theology Energetics
B 	Physics Atomic physics Tools of physics Astrophysics	Musicology Musical forms Theories of music Musical instruments	Psychology Psychological theories Psychometrics Defence mechanism	Energetics Alternative sources Heating techniques Oil and gas companies
C 	Physics Astrophysics Physics of stars Giants	Musicology Musical forms Dance forms Ballet	Psychology Psychological theories Communication Conflict	Energetics Heating techniques Heat engines Fuel to heat engines

5.2 Clustering Algorithm

The nested clusters have been obtained using OPTICS algorithm [20]. This is a well-known clustering algorithm that analyses local object densities and present them on the so-called *reachability plot* (fig. 10a). In the reachability plot "valley" regions indicate natural clusters found in the dataset; the deeper they are, the more dense are the clusters they designate. The height of the bar (or bars)

between valleys show how clusters are separated from each other. Subclusters of a parent cluster are indicated by deeper and narrower valleys (or "dents") found in the bottom of the "parent" valley. Nested clustering of a dataset can be then obtained from the reachability plot by finding and analysing it's valleys.

There are a few methods of extracting clusters from reachability plot. The most well-known are:

ζ -clustering [20] recognizes clusters on the basis of down and up steep areas which slope is defined by a parameter ζ . A cluster starts in a steep down area and ends in a steep up area in such positions that the start and end points have approximately the same reachability value. A similar idea is used in the *gradient*-clustering [21]. Here, the algorithm finds *inflexion points*, i.e. points where the gradient (the difference between reachability values of adjacent points) changes significantly. This significance is set by a parameter t . The results given by these two methods usually form complex structures with many levels of subclusters (fig. 10b and 10c) and, in most real data cases, are highly sensitive to their parameters.

Tree-clustering [22] finds the most significant local maxima in the reachability plot. It sorts them decreasingly and treats as split points that divide higher level clusters into smaller ones. This method is relatively insensitive to it's parameter *significance*, but it has been shown [21], that some nested clusters cannot be found using local maxima identification (fig. 10d).

SCI – Simple Cluster Identification algorithm [23], created to achieve best purity of clusters regardless of the coverage (the percentage of objects that are clustered to any cluster), works based on a simplification that the cluster is a maximal sequence of points which have comparable reachability values. This method returns only "flat" clusters. The nested clusters are not obtained here directly, but by repeating this procedure within each of the clusters separately.

5.3 Experiments and Results

In our experiments we compare the application of methods presented in subsection 5.2 to find the best one for extracting the clusters of Wikipedia articles compared to human-made categories. We also test the sensitivity to input parameters of these methods (i.e. stability of results while changing ζ , t and *significance*).

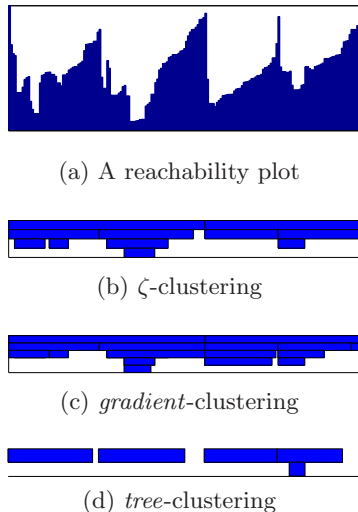


Fig. 10: An exemplary real data reachability plot and structures of nested clusters extracted using different methods.

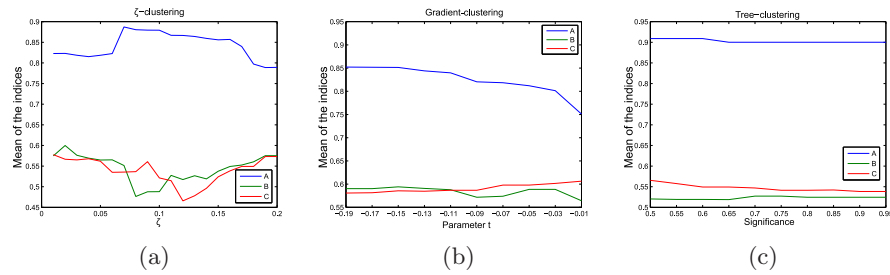


Fig. 11: The averaged results of external clustering validation indices for test datasets ($hRand_{ca}$, hI_{ca}^* and hF -measure) for ζ - (a), gradient- (b) and tree- (c) clustering algorithms.

Figure 11 shows that the *tree*-clustering method returns more stable results than the other 2 techniques. This method gives also less complex clustering structures (this can be shown in the figure 10d) and exhibits the best performance for type A datasets (which are datasets without subclasses). However, the other 2 ones, when their parameters are properly set, work better for B and C datasets, i.e. in more complicated, nested cases.

Figure 12 shows the averaged results of nested clusters extraction for each of the methods with optimal settings. We also provide the results for mixing the datasets together (for example datasets B1 and B2 together, or datasets A3, B3 and C3 together etc.) to test how OPTICS works with bigger datasets with more complex trees of categories.

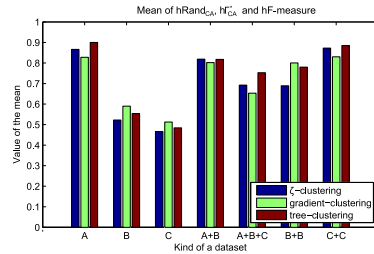


Fig. 12: Clusters extraction algorithms performance.

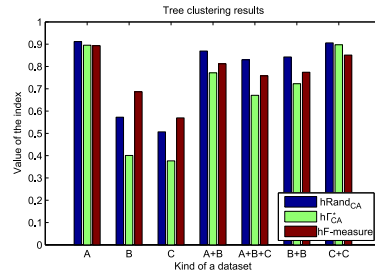


Fig. 13: 3 external validation indices for results of tree-clustering.

It is obvious the clustering of datasets of type B and C are much less accurate than that of type A, because the structure is more complicated. What is surprising is that the clustering of more complex datasets, namely B+B, A+B, A+B+C and C+C, provide better results than those of only B or C. It is because of the fact the cluster extraction algorithms use only relative knowledge about

the distances between clusters in the feature space. For datasets of type B or C, where all articles are from the same parent category, they should be placed in the same cluster and then divided into subclusters. However, clustering algorithms always try to find the most distinguishable groups of objects in the dataset and place them into different clusters. In the case when all objects are from the same category, algorithms must split this category into different clusters causing lower values of the external validation indices. When a dataset contains articles from other categories (like B+B examples), then the main classes are correctly clustered and this results in significantly higher values of the overall external validation indices.

In figure 13 we show the comparison of the evaluation of the *tree*-clustering algorithm using $hRand_{ca}$, $h\Gamma_{ca}^*$ and $hF - measure$. It can be seen that in this case, they all measure the same tendencies: results with little differences in dynamics, range of the returning values with $h\Gamma_{ca}^*$ as most dynamically changing. The impact of advantages and disadvantages of these indices, discussed in the previous section, is not clearly seen here.

6 Conclusions and Future Work

The main contribution of this paper is an examination of existing and a development of some new external validation methods applied to nested clustering. This validation step is crucial in researching good tools for content oriented organization of text documents.

We added two requirements to those of Kiritchenko et al., which should be fulfilled by a satisfactory nested clustering measure. Then, we proposed such measures: we adopted *Rand* and Hubert's Γ^* indices by calculating them on non-binary similarity matrices. We proposed how to get such non-binary *CA*-type similarity matrices. After evaluation of proposed measures, we employed the most promising of them to judge clusterings of Wikipedia articles with different variants of OPTICS algorithm.

Measure evaluation part shows that $hRand_{ca}$, $h\Gamma_{ca}^*$ and $hF-measure$ are the most promising indices of nested clustering quality wrt. external reference. Results presented in the practical part demonstrates that OPTICS can be used for text documents clustering. Depending on whether we want more complex, or simpler, cluster trees, *gradient*-, ζ - or *tree*-clustering extraction algorithm can be employed.

Having established validation measures, our future work will be focused on the development of text representations that will be able to capture semantics of the text. Some promising ideas for this goal is the use of semantic correlations between words obtained from an external lexical knowledge like Wordnet [24] or/and, in the case of on-line articles, the exploitation of the information carried by links between articles.

Acknowledgment This work has been supported by the National Centre for Research and Development (NCBiR) under research Grant No. SP/I/1/77065/1

SYNAT: "Establishment of the universal, open, hosting and communication, repository platform for network resources of knowledge to be used by science, education and open knowledge society".

References

1. Manning C.D., Raghavan P., S.H.: An Introduction to Information Retrieval. Cambridge University Press (2008)
2. D., G.A.: A review of hierarchical classification. (1987) 119–137
3. Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., Dougherty, E.R.: Model-based evaluation of clustering validation measures. *Pattern Recognition* **40** (2007) 807 – 824
4. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. *Journal of Intelligent Information Systems* **17** (2001) 107–145
5. Halkidi, M., Vazirgiannis, M.: A density-based cluster validity approach using multi-representatives. *Pattern Recognition Letters* **29** (2008) 773 – 786
6. Milligan, G., Cooper, M.: An examination of procedures for determining the number of clusters in a data set. *Psychometrika* **50** (1985) 159–179
7. Dimitriadou, E., Dolniar, S., Weingessel, A.: An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika* **67** (2002) 137–159 10.1007/BF02294713.
8. Menardi, G.: Density-based silhouette diagnostics for clustering methods. *Statistics and Computing* (2010) 1–14 10.1007/s11222-010-9169-0.
9. Zalik, K.R., Zalik, B.: Validity index for clusters of different sizes and densities. *Pattern Recognition Letters* **32** (2011) 221 – 234
10. Aliguliyev, R.M.: Performance evaluation of density-based clustering methods. *Information Sciences* **179** (2009) 3583–3602
11. M., R.W.: Objective criteria for the evaluation of clustering method. *Journal of the American Statistical Association* **66** (1971) 846–850
12. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* **2** (1985) 193–218
13. Milligan, G.W., C., S.S., Sokol, L.M.: The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1983)
14. Kiritchenko S., Matwin S., N.R.F.F.: Learning and evaluation in the presence of class hierarchies: Application to text categorization. *Lecture Notes in Artificial Intelligence* **4013** (2006) 395–406
15. James, R.F.: Methods of comparing classifications. *Annual Review of Ecology and Systematics* **5** (1974) 101–113
16. Raghavan, V.V., Ip, M.Y.L.: Techniques for measuring the stability of clustering: a comparative study. In: *Proceedings of the 5th annual ACM conference on Research and development in information retrieval. SIGIR '82, New York, NY, USA, Springer-Verlag New York, Inc. (1982) 209–237*
17. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Communications of the ACM* **18** (1975) 613–620
18. S.M., W., N., I.: *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann (1997)
19. I.T., J.: *Principal Component Analysis*. Springer (2002)

20. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. *SIGMOD Rec.* **28** (1999) 49–60
21. Brecheisen, S., Kriegel, H.P., Krger, P., Pfeifle, M.: Visually mining through cluster hierarchies. In: *Proceedings of the Fourth SIAM International Conference on Data Mining*. (2004)
22. Sander, J., Qin, X., Lu, Z., Niu, N., Kovarsky, A.: Automatic extraction of clusters from hierarchical clustering representations. In Whang, K.Y., Jeon, J., Shim, K., Srivastava, J., eds.: *Advances in Knowledge Discovery and Data Mining*. Volume 2637 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg (2003) 567–567
23. P, D., Roy, S.: Optics on text data: Experiments and test results. In: *Proceedings of the 2006 SIAM International Conference on Data Mining*. (2006)
24. Ch., F., ed.: *WordNet. An Electronic Lexical Database*. The MIT Press (1998)