STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXI**, Number 1, 2016

DISCOVERING PATTERNS IN DATA USING ORDINAL DATA ANALYSIS

ADRIANA M. COROIU, RADU D. GĂCEANU, AND HORIA F. POP*

Abstract. Discovering patterns in data is becoming more and more important for different fields of research. The analysis of ordinal data is sensitive and requires special attention. In order to analyze ordinal data, we may use various criteria. In our paper, we present a solution by using different linkage criteria (ward, median, centroid, weighted, complete and single linkage method) with agglomerative clustering algorithms.

To evaluate and interpret our results we have considered some internal and external evaluation indexes for clustering (also known as cluster analysis). The experiments reveal different comparative results. To validate our clustering results, we used pair-counting measures (Jaccard, Recall, Rand and Fowlkes-Mallows indexes), BCubed-based measures (F1-Measure), set-matching-based measures and editing-distance measures (Purity, Precision and Recall) for external evaluation and Silhouette index for analyzing intrinsic characteristics of a clustering (internal evaluation).

The comparative experiments for different linkage methods suggest that for an ordinal data set, by using ward linkage methods we achieve more accurate results in terms of cluster validity than others linkage criteria applied to our data set.

1. Introduction

Clustering is one of the most useful methods to discover patterns in data [21]. Due to its role related to discover structures in data, we can say that clustering is a good tool for exploration of data. From the early ideas of the 1930s [12], this field has experienced a vast extension filed by new concepts and computational difficulties [1]. Nowadays, the omnipresence of clustering in our life is overwhelming.

Received by the editors: November 20, 2015.

2010 *Mathematics Subject Classification.* 68T10, 62H30, 62-07.

1998 *CR Categories and Descriptors.* I.5.3 [**Pattern Recognition**]: Clustering – *Algorithms – Data analysis*.

Key words and phrases. Ordinal Data Analysis, Agglomerative Clustering.

* Corresponding author.

Comprehending information has turned into a basic goal of intelligent data analysis (IDA), data mining (DM), sensor fusion, image comprehension, and logic-driven system modeling. Clustering has turned into an equivalent word in a differentiated set of philosophies and algorithms that are almost exclusively data-driven and in which any optimization is predominantly, if not exclusively, data-oriented.

Clustering offers ascend to an assortment of data granules whose utilization uncovers the structure of information. Indeed, to a short and unsophisticated search of the web for a simple search of any library database returns thousands of hits, revealing an impressive breadth of applications: from bio-medicine to marketing, engineering, economics, biological sciences, chemistry, military, food engineering, finance, and education [13].

In literature, there are different data analysis methods targeting continuous data, but there are few methods for ordinal data analysis. The analysis of ordinal data is more sensitive because we cannot apply the usual formulas, such as mean, or standard deviation on such data.

Our aim is to study the applicability of clustering algorithms on ordinal data. We consider a Naive agglomerative clustering approach [12] and the Slink algorithm [26] with several linkages and we apply them on a standard data set [16] with ordinal data.

This paper is structured as follows: Section 2 presents the main idea on which this paper is based. Section 3 and Section 4 present important notions related to clustering, data types, linkage methods and the agglomerative clustering algorithms that are used in the paper. Section 5 describes the performed experiments and provides an interpretation of the obtained results, and finally, the last section draws the conclusions and presents ideas for future work.

2. Motivation

According to the scientific literature [12, 28], clustering is one of the most popular method of extracting essential information from data. Ordinal data is a particular type of data, and due to its properties, is very sensitive and require different techniques of analysis.

Ordinal data is a type of data gathered from different surveys of social sciences such as psychology, education, sociology, medicine. In these domains, the researchers are using different questionnaires in order to gather the information from the patients or users. The analysis of data from these domains was the main point of start in this paper. The information that can be collected using clustering may offer precious information for a therapist, psychologist or sociologist.

One of the main issues of ordinal variables is that distances, means, or standard deviations cannot be directly computed. Even if ranks are associated to the given categories, the size of the difference between two categories is in general inconsistent. If the difference between categories was measurable then the variables would be considered interval-based or ratio-based in which case distances, mean, and standard deviation would be well-defined. Since this is not always the case, the problem of ordinal data analysis is important particularity in fields like economics or social-behavioural sciences, where data is often ordinal by nature [4].

3. Theoretical background

Unlike classification, which breaks down class-labeled data sets, clustering investigates data without class labels. The objects are clustered together based on maximizing the intraclass similarity and minimizing the interclass similarity [2]. Clustering is the main unsupervised learning method. The learning procedure is unsupervised since the information, samples are not class labeled.

We introduce some fundamental notions related to clustering: types of data, distance, and similarity measures.

3.1. **Challenges in clustering.** Clustering is a challenging research field. There are some requirements for clustering as a data mining tool, as well as aspects that can be used for comparing clustering methods. The following features should be considered:

- Scalability: Many clustering algorithms deal with small data sets containing fewer than several hundred data objects; but nowadays, a large database may contain millions or even billions of objects. Clustering on only a sample of a given large data set may conduct to biased results. In this case, highly scalable clustering algorithms are needed [10].
- Ability to deal with different types of attributes: Many algorithms are designed to cluster numeric (interval-based) data. However, applications may require clustering other data types, such as binary, nominal (categorical), and ordinal data, or mixtures of these data types. Recently, more and more applications need clustering techniques for complex data types such as graphs, sequences, images, and documents.
- Discovery of clusters with arbitrary shape: Numerous clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. Consider sensors, for example, which are often deployed for environmental surveillance.

- Requirements for domain knowledge to determine input parameters: Many clustering algorithms require users to provide domain knowledge in the form of input parameters such as the desired number of clusters. Consequently, the clustering results may be sensitive to such parameters. Parameters are often hard to determine, especially for high-dimensionality data sets and where users have yet to grasp a deep understanding of their data. Requiring the specification of domain knowledge not only burdens users, but also makes the quality of clustering difficult to control.
- Ability to deal with noisy data: Most data sets contain exceptions and/or missing, obscure, or mistaken information [10]. Clustering algorithms can be sensitive to such noise and may produce poor-quality clusters. Therefore, we need clustering methods that are robust to noise.
- Incremental clustering and insensitivity to input order: In many applications, incremental updates (newer data) may arrive at any time. Some clustering algorithms cannot incorporate incremental updates into existing clustering structures and, instead, have to recompute a new clustering from scratch. Clustering algorithms may also be sensitive to the input data order. That is, given a set of data objects, clustering algorithms may return dramatically different clustering depending on the order in which the objects are presented. Incremental clustering algorithms and algorithms that are insensitive to the input order are needed [11].
- Capability of clustering high-dimensional data: An information set can contain various measurements or qualities. When clustering documents, for instance, every keyword can be viewed as a measurement, and there are regularly a huge number of keywords. Most clustering algorithms are great at taking care of low-dimensional data, for example, data sets including just a few measurements. Discovering clusters of data in a high dimensional space represents a challenge, particularly considering that such data can be exceptionally inadequate and very skewed.
- Constraint-based clustering: Real-world applications may need to perform grouping under different sorts of limitations. A challenging task is to find data groups with good clustering behavior that satisfy specified constraints.
- Interpretability and usability: Users need clustering results to be interpretable, understandable, and usable.

3.2. **Types of data – Data objects and attributes.** Data sets consist of data objects. Data objects are generally described by attributes or variables. Data objects are also known as samples, examples, instances, data points, or objects. If the data objects are stored in a database, they are called data tuples. That is, the rows of a table correspond to the data objects, and the columns correspond to the attributes. In the following section, we will have a short description of these notions.

The world encompassing us creates different sorts of data. The formal representation and association of patterns mirror the path in which we intend to process the data. The most broad scientific taxonomy being in common use distinguishes among numeric, ordinal, and nominal variables [10].

An attribute is a data field, represented as a characteristic or as a feature of a data object. The nouns attribute, dimension, feature, and variable are often used interchangeably in the literature. The term dimension is commonly used in data warehousing. Machine learning literature tends to use the term feature, while statisticians prefer the term variable. Data mining commonly uses the term attribute. Observed values for a given attribute are known as observations. A set of attributes used to describe a given object is called an attribute (feature) vector.

The type of an attribute is determined by the set of possible values: nominal, binary, ordinal, or numeric. In the following subsections, we offer a short presentation for each type.

The values of a nominal attribute are symbols or names of things. Each value represents some kind of category, code, or state, therefore the nominal attributes are also referred to as categorical. The values do not have any meaningful order. Because nominal attribute values do not have any meaningful order about them and are not quantitative, it makes no sense to find the mean or median value for such an attribute, given a set of objects.

A binary attribute is a nominal attribute with only two categories or states: 0 or 1, where 0 typically means that the attribute is absent, and 1 means that it is present. Binary attributes are referred to as Boolean if the two states correspond to true and false. A binary attribute is symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1.

An ordinal attribute is an attribute with possible values with a meaningful order or ranking, but the difference between successive values is not known. Ordinal attributes are useful for registering subjective assessments of qualities that cannot be measured objectively; thus ordinal attributes are often used in surveys for ratings. They may also be obtained from the discretisation of numeric quantities by splitting the value range into a finite number of ordered categories [18].

The nominal, binary, and ordinal attributes are qualitative and these describe a feature of an object without giving an actual size or quantity. The values of qualitative attributes are typically words representing categories. If integers are used, they represent computer codes for the categories, as opposed to measurable quantities.

A numeric attribute is quantitative, in other words, a measurable quantity, represented in integer or real values. Numeric attributes can be interval-scaled or ratio-scaled.

Interval-scaled attributes are measured on a scale of equal-size units. The values of interval-scaled attributes have orders and can be positive, 0, or negative. Thus, in addition to providing a ranking of values, such attributes allow us to compare and quantify the difference between values.

A ratio-scaled attribute is a numeric attribute with an inherent zero-point. That is, if a measurement is the ratio-scaled, we can speak of a value as being a multiple (or ratio) of another value. In addition, the values are ordered, and we can also compute the difference between values, as well as the mean, median, and mode.

The concept of distance is the essential component of any form of clustering that helps us navigate through the data space and form clusters. By computing dissimilarity, we can sense and articulate how close together two patterns are and, based on this closeness, allocate them to the same cluster. In the case of continuous features there is a long list of distance functions. Each of these functions implies a different view of the data because of their geometry [3].

4. Clustering algorithms

Clustering algorithms partition the objects into groups, or clusters, so that objects within a cluster are similar to one another and dissimilar to objects in other clusters. Similarity is commonly defined in terms of how close the objects are in space, based on a distance function. The quality of a cluster may be represented, for example, by its diameter, the maximum distance between any two objects in the cluster.

Clustering can be used as a standalone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, and to focus on a particular set of clusters for further analysis. Alternatively, it may serve as a preprocessing step for other algorithms, such as characterization, attribute subset selection, and classification, which would then operate on the detected clusters and the selected attributes or features [6, 8].

4.1. **Partitioning methods.** Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and $k <= n$. That is, it divides the data into k groups such that each group

must contain at least one object [14]. In other words, partitioning methods, conduct a one-level partitioning on data sets. The basic partitioning methods typically adopt exclusive cluster separation. That is, each object must belong to exactly one group. While partitioning methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups, in some situations we may want to partition our data into groups at different levels such as in a hierarchy.

4.2. **Hierarchical methods.** A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed.

The agglomerative approach starts with each object forming a separate group. It successively merges the objects or groups close to one another, until all the groups are merged into one, or a termination condition holds. The divisive approach, starts with all the objects in the same cluster. In each successive iteration, a cluster is split into smaller clusters, until eventually each object is in one cluster, or a termination condition holds.

Hierarchical clustering methods can be distancing, biased or density, and continuity based. Various extensions of hierarchical methods consider clustering in subspaces as well [25]. An agglomerative hierarchical clustering method uses a bottom-up strategy. It typically starts by letting each object form its own cluster and iteratively merges clusters into larger and larger clusters, until all the objects are in a single cluster or certain termination conditions are satisfied. The single cluster becomes the hierarchys root.

For the merging step, it finds the two clusters that are closest to each other (according to a similarity measure), and combines them to form one cluster. Because two clusters are merged per iteration, where each cluster contains at least one object, an agglomerative method requires at most $n - 1$ iterations [18].

In the bottom-up mode known as an agglomerative approach, we treat each pattern as a single-element cluster and then successively merge the closest clusters. At each pass of the algorithm, we merge the two closest clusters. The process is repeated until we get to a single data set or reach a certain predefined threshold value.

The top-down approach works in the opposite direction: we start with the entire set treated as a single cluster and keep splitting it into smaller clusters. Intuitively, we can easily envision three typical ways of computing the distance between the two clusters.

4.3. **Linkage methods used in clustering.** One advantage of hierarchical clustering algorithms over partitioning algorithms is that the number of clusters does not need to be known in advance, there is no initial assignment of data to clusters needed, and also a distance measure between data items is not necessary. The algorithms only need an intercluster similarity measure (which may be based on a data point similarity measure). The most common linkage measures are [17]:

- single linkage;
- complete linkage;
- average linkage;
- median linkage method;
- weighted linkage method;
- centroid linkage method;
- ward linkage method.

In the following subsection all these linkage methods are described.

We have the next notation to describe the linkages used by the various methods: Cluster r is formed from clusters p and q.

- n_r is the number of objects in cluster r;
- x_{ri} is the i-th object in cluster r.

The single linkage or minimum distance rule starts out by finding the two points with the minimum distance. These are placed in the first cluster. At the next stage a third point joined the already-formed cluster of two if the minimum distance to any of the members of the cluster is smaller than the distance between the two closest unclustered points. Otherwise, the two closest unclustered points are placed in a cluster. The process continues until all points end up in one cluster. The distance between two clusters is defined as the shortest distance from a point in the first cluster that is closest to a point in the second [26]. This linkage method is also called nearest neighbor:

$$d(r,s) = min(dist(x_{ri}, x_{sj})), i \in (i, ..., n_r), j \in (1, ..., n_s) \quad (1)$$

The complete linkage option starts out in just the same way by clustering the two points with the minimum distance. However, the criterion for joining points to clusters or clusters to clusters involves the maximum (rather than minimum) distance. That is, a third point joined the already formed cluster if the maximum distance to any of the members of the cluster is smaller than the distance between the two closest unclustered points. In other words, the distance between two clusters is the longest distance from a point in the first cluster to a point in the second cluster. This linkage method is also called furthest neighbor [27]:

$$d(r,s) = max(dist(x_{ri}, x_{sj})), i \in (i, ..., n_r), j \in (1, ..., n_s) \quad (2)$$

The average linkage option starts out in the same way as the other two. However, in this case the distance between two clusters is the average distance from points in the first cluster to points in the second cluster. The average linkage uses the average distance between all pairs of objects in any two clusters:

$$d(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_r} dist(x_{ri}, x_{sj}) \quad (3)$$

The median linkage uses the euclidean distance between weighted centroids of the two clusters:

$$d(r,s) = ||\widetilde{x_r}\widetilde{x_s}||_2 \quad (4)$$

where $\widetilde{x_r}$ and $\widetilde{x_s}$ are weighted centroids for the clusters r and s. If cluster r was created by combining clusters p and q, $\widetilde{x_r}$ is defined recursively as: $\widetilde{x_r} = \frac{1}{2}(\widetilde{x_p} + \widetilde{x_q})$ (5).

The weighted average linkage uses a recursive definition for the distance between two clusters.

If cluster r was created by combining clusters p and q, the distance between r and another cluster s is defined as the average of the distance between p and s and the distance between q and s:

$$d(r,s) = \frac{d(p,s)+d(q,s)}{2} \quad (6)$$

Centroid linkage uses the euclidean distance between the centroids of the two clusters:

$$d(r,s) = ||\overline{x_r} - \overline{x_s}||_2 \quad (7)$$

where $\overline{x_r} = \frac{1}{n_r} \sum_{i=1}^{n_r} x_{ri}$.

Wards linkage method starts out by finding two points with the minimum within groups sum of squares. The points continue to be joined to the first cluster or to other points depending on which combination minimizes the error sum of squares from the group centroid. This method is also known as a k-means approach. Closely related to the Wards algorithm is the Howard-Harris algorithm. The Howard-Harris algorithm is a hierarchical divisive method which uses the k-means method of assigning cases to the clusters [27].

Ward's linkage uses the incremental sum of squares, that is, the increase in the total within-cluster sum of squares as a result of joining two clusters. The within-cluster sum of squares is defined as the sum of the squares of the distances between all objects in the cluster and the centroid of the cluster. The sum of squares measure is equivalent to the following distance measure $d(r,s)$:

$$d(r,s) = \sqrt{\frac{2n_r n_s}{(n_r+n_s)}} ||\overline{x_r} - \overline{x_s}||_2 \quad (8)$$

where $\overline{x_r}$ and $\overline{x_s}$ are the centroids of clusters r and s, n_r and n_s are the number of elements in clusters r and s. In some references the Ward linkage does not use the factor of 2 multiplying n_r and n_s.

4.4. **Cluster validity.** In order to establish the quality of the results gathered in a clustering, we can analyze the external validation indexes and the internal validation indexes [19].

- External validation. In external validation, the quality of the algorithm is evaluated by comparing the resulting clusters with pre-specified information. There are many external validation measures like Purity, Rand, Entropy, Jaccard coefficient or Fowlkes-Mallows Index (FM) [7]. The clusters formed are evaluated and interpreted according to the distance between data points and cluster centers of each cluster [15].
- Internal validation. For internal validation, the evaluation of the resulting clusters is based on the clusters themselves, without additional information or repeating of the clustering process. This family of techniques is based on the assumption that the algorithms should search for clusters whose members are close to each other and far from members of other clusters.

5. Computational experiments

Clustering is the process of partitioning a set of data objects (or observations) into subsets. Each subset is a cluster, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. In this context, different clustering methods may generate different clustering tree on the same data set. Hence, clustering is useful in that it can lead to the discovery of previously unknown groups within the data.

Hierarchical clustering algorithms may be identified as either hierarchical agglomerative or hierarchical divisive, meaning that they contract or expand the space between groups of points in the multivariate space. Wards method and complete linkage rules are of the divisive variety and tend to create clusters of roughly equal size that are hyper-spherical in form. The average linkage method neither expands nor contracts the original space, while the single linkage tends to agglomerate or contract the space between groups of points in multivariate space.

The experiments reveal different results. Based on them, we provide comparisons. To validate our clustering results, we have the following measures, pair counting measures, BCubed based measures, set matching based measures and editing distance measures - for external evaluation and Silhouette index for analyzing intrinsic characteristics of a clustering resulted structure - internal evaluation.

For all these measures of validation in our experiments we gathered values of the specified index of them. For pair counting measures, we have values

for the following indexes: Jaccard, Recall, Rand and Fowlkes-Mallows indexes and for set matching based measures and editing distance measures we have values for Purity, Precisions and Recall.

The Jaccard index is a common index for binary (and non-binary) variables [17].

If $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$ are two vectors with all real and positive x_i, y_i, then the Jaccard similarity coefficient is defined as:

$$J(x, y) = \frac{\sum_i min(x_i, y_i)}{\sum_i max(x_i, y_i)} \tag{9}$$

The Rand index in statistics, and in particular in data clustering, is a measure of the similarity between two data clustering trees. Rand index is related to the accuracy, but is applicable even when class labels are not used [23].

For a set of n elements $S = (o_1, ..., o_n)$ and two partitions of S to compare, $X = (X_1, ..., X_r)$, a partition of S into r subsets, and $Y = (Y_1, ...Y_s)$, a partition of S into s subsets, and with the definition of the following: a – the number of pairs of elements in S that are in the same subset in X and in the same subset in Y; b – the number of pairs of elements in S that are in different subsets in X and in different subsets in Y; c – the number of pairs of elements in S that are in the same subset in X and in different subsets in Y; d – the number of pairs of elements in S that are in different subsets in X and in the same subset in Y, we have:

$$R = \frac{a+b}{a+b+c+d} \tag{10}$$

For Fowlkes-Mallows index we have the following [7]: two clustering trees of n objects identified A_1 and A_2. The trees A_1 and A_2 can be cut to produce $k \in \{2, \ldots, n-1\}$ clusters for each tree. And for each value of k, we have: $M = [m_{i,j}], i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, k\}$ where $m_{i,j}$ is of objects common between the $i-$th cluster of A_1 and $j-$th cluster of A_2. Fowlkes-Mallows index (B_k) can then be computed for every value of k and we have $0 \leq B_k \leq 1$. For a specific value of k we have the formula (11):

$$B_k = \frac{T_k}{\sqrt{P_k Q_k}} \tag{11}$$

$$T_k = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{i,j}^2 - n \tag{12}$$

$$P_k = \sum_{i=1}^{k} (\sum_{j=1}^{k} m_{i,j})^2 - n \tag{13}$$

$$Q_k = \sum_{j=1}^{k} (\sum_{i=1}^{k} m_{i,j})^2 - n \tag{14}$$

Fowlkes-Mallows index showed that on using two unrelated clustering trees, the value of this index approaches zero as the number of total data points chosen for clustering increase; whereas the value for the Rand index for the same data quickly approaches making Fowlkes-Mallows index a much accurate representation for unrelated data [7].

In order to compute the purity of a set of clusters, first, we calculate the purity for each cluster, according to formula (15):

$$P_j = \frac{1}{n_j} Max(n_j^i) \tag{15}$$

In other words, P_j is a fraction of the overall cluster size that the largest class of objects assigned to that cluster represents. The overall purity of the clustering solution is obtained as a weighted sum of the individual cluster purities and given as:

$$Purity = \sum_{j=1}^{m} \frac{n_j}{n} P_j \tag{16}$$

Were n_j is the size of cluster j, m is the number of clusters, and n is the total number of objects [15].

For computations of the recall and precision indexes we have the following formulas [9].

$$Recall(i,j) = \frac{n_{ij}}{n_i} \tag{17}$$

$$Precision(i,j) = \frac{n_{ij}}{n_j} \tag{18}$$

Were n_{ij} is the number of objects of class i that are in cluster j, n_j is the number of objects in cluster j and n_i is the number of objects in class i.

The silhouette is the average, over all clusters, of the silhouette width of their points. If x is a point in the cluster Ck and nk is the number of points in Ck, then the silhouette width of x is defined by the ratio where $a(x)$ is the average distance between x and all other points in Ck, and $b(x)$ is the minimum of the average distances between x and the points in the other clusters [20].

For a given point x, its silhouette width ranges from 1 to 1. The higher the silhouette, the more compact and separated are the clusters [24].

The Silhouette coefficient is an example of such an evaluation, where a higher Silhouette coefficient score relates to a model with better defined clusters. The Silhouette coefficient is defined for each sample and is composed of two scores [24]: a – the mean distance between a sample and all other points in the same class and b – the mean distance between a sample and all other points in the next nearest cluster.

The Silhouette coefficient s is given as:

$$s = \frac{b-a}{max(a,b)} \tag{19}$$

First data set used in our experiment was the Dermatology data set [16] — a data set with 366 instances and 33 attributes and the attribute characteristics are categorical. In this data set, every feature (clinical and histopathological) was given a degree in the range of 0 to 3. A value of 0 indicates that the feature was not present, 3 indicates the largest amount possible, and a value of 1 or 2 indicates the relative intermediate values.

The second data set used in our experiment was Chronic Kidney Disease data set [16]. The original data set Chronic Kidney Disease is a data set

with 400 instances and 24 attributes divided in 11 numeric attributes and 13 nominal attributes. In order to have an ordinal data set, we create a subset of this original data set. To get the ordinal subset for Chronic Kidney Disease Data Set, we have performed the following: we have deleted the nominal attributes and we process the numeric attributes. The 11 remaining attributes are: blood pressure, albumin, blood glucose, blood urea, serum creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count and red blood cell count. We scaled these numerical values and we transformed the numerical values in three ordinal values: small (coded with 1), medium (coded with 2) and high (coded with 3).

For our experiment, we have used an open source data mining software written in Java called ELKI [5]. We have performed experiments with two agglomerative clustering algorithms: Slink and Naive agglomerative clustering algorithm. For these two algorithms we have chosen different linkage criteria (ward, median, centroid, weighted, complete, average and single linkage).

In order to evaluate our results, we have used several measures for internal and external evaluation of a clustering. For external evaluation, we have chosen the following methods of evaluation: pair counting measures, BCubed based measures, set matching based measures, editing distance measures.

Pair counting measures are an approach based on counting pairs of objects that are classified in the same way in both clustering trees. For Pair counting measures we have gathered the following values, indexes: Jaccard, Recall, Rand and Fowlkes-Mallows indexes.

The Jaccard index has values between 0 – independent clustering tree and 1 - identical clustering tree. The Rand index is an index which correctly classifies pairs of elements, its value between 0 and 1; he value 1 means that two partitions perfectly agree. The FM is an external evaluation method that is used to determine the similarity between two clusters. A higher value for Fowlkes-Mallows index indicates a greater similarity between two clusters [22].

Our experiments show a greater value of Jaccard, Rand and Fowlkes-Mallows indexes if we have applied ward linkage method in comparisons to other linkage method analyzed (median, centroid, weighted, complete, average and single linkage). Consequently, ward linkage method provides an accurate clustering than another linkage methods applied in an agglomerative clustering algorithm. The second evaluation methods on which we have values of indexes is BCubed-based measures – BCubed metrics decompose the evaluation process estimating the precision and recall associated with each item in the distribution. The item precision represents how many items in the same cluster belong to its category. At the same time, the recall associated with one item represents how many items from its category appear in its cluster. BCubed metrics independently compute the precision and recall associated to

Index/Linkage Method	Ward		Median		Centroid		Complete		Single	
(Algorithm) for 2 data sets	Nave	Slink	Nave	Slink	Nave	Slink	Nave	Slink	Nave	Slink
Jaccard	0.19	0.65	0.23	0.27	0.51	0.52	0.19	0.27	0.19	0.60
	0.23	0.67	0.26	0.31	0.55	0.55	0.22	0.32	0.22	0.64
Recall	0.19	0.65	0.23	0.27	0.51	0.52	0.19	0.27	0.19	0.75
	0.23	0.67	0.26	0.31	0.55	0.55	0.22	0.32	0.22	0.75
Rand	0.21	0.65	0.44	0.47	0.63	0.64	0.21	0.29	0.41	0.90
	0.24	0.67	0.45	0.48	0.68	0.67	0.25	0.35	0.47	0.91
FowlkesMallows	0.44	0.81	0.48	0.52	0.71	0.72	0.43	0.52	0.43	0.75
	0.47	0.83	0.49	0.55	0.73	0.74	0.47	0.55	0.48	0.77
F1-Measure	0.45	0.87	0.53	0.56	0.71	0.72	0.47	0.57	0.50	0.83
	0.50	0.89	0.55	0.60	0.73	0.74	0.50	0.59	0.53	0.88
Purity	0.29	0.78	0.39	0.43	0.61	0.62	0.32	0.41	0.36	0.84
	0.30	0.79	0.41	0.46	0.66	0.63	0.35	0.44	0.38	0.88
Precision	0.98	0.97	0.84	0.85	0.85	0.86	0.97	0.97	0.83	0.83
	0.97	0.96	0.83	0.84	0.83	0.84	0.95	0.96	0.80	0.81
Silhouette	0.63	0.35	0.69	0.67	0.63	0.62	0.62	0.60	0.70	0.47
	0.64	0.40	0.71	0.69	0.64	0.64	0.64	0.62	0.78	0.52

TABLE 1. Values achieved. Columns represent achieved index for both algorithms in different linkage methods

each item in the distribution. The precision of an item represents the amount of items in the same cluster that belong to its category. The recall of an item represents how many items from its category appear in its cluster. In our case, our experiments outline a higher value of precision which means that the cluster achieved is more precision-oriented. A higher value of precision is achieved applied no matter of the linkage methods on agglomerative algorithms.

We can reach a maximum value for inverse purity by making a single cluster with all items. In our experiments, the results show a lower value of inverse purity and a higher value of purity, which means that the clustering methods return good results. The results of the purity achieved by applying our linking criteria shows that if we chose the single linkage method, for purity we received a higher value (0.84 and 0.88) than if we have been chosen ward linkage method (value of purity 0.78 and 0.79), so according to this measurement set matching, the single linkage criteria provides better results than other linkage criteria.

For the internal evaluation of a clustering, the analyzed and achieved value is the silhouette index, which validates the clustering performance based on the pairwise difference between cluster distances. In our experiments, we have obtained a higher value with ward linkage methods than with single linkage methods. The values are between 0.50 and 0.70 and these outline that reasonable structure has been found.

Table 1 shows the results of the two algorithms using different linkage criteria, using two ordinal data sets.

6. Conclusion and future work

The aim of this paper was to study the agglomerative clustering algorithms in the case of an ordinal data sets. We have performed tests using Naive algorithm and Slink agglomerative clustering algorithm. We have studied the appropriate linkage criteria to be used for particular algorithms. Our experiments reveal good results in term of clustering evaluation for the ward linkage criteria as compared to other linkage criteria.

One of the future point of our research is related, first to outlier detection for ordinal data and, second, to the knowledge based clustering and as well as figuring out a way to introduce robustness via fuzzy logic.

Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNDI–UEFISCDI, project number PN-II-PT-PCCA-2011-3.2-0917.

References

[1] E. Backer, A. Jain, *A clustering performance measure based on fuzzy set decomposition*, IEEE Trans. Pattern Anal. Mach. Intell., vol.PAMI-3, no. 1, (1981), 6675.
[2] C. Charu, *Data Classification: Algorithms and Applications*, CRC Press, (2014).
[3] V. Cherkassky F. Mulier, *Learning From Data*, Concepts, Theory,and Methods, Wiley, (1998), 23-26.
[4] A. M. Coroiu, R. D. Găceanu, H. F. Pop, *Ordinal data analysis using agglomerative clustering algorithms*, In Knowledge Engineering Principles and Techniques, Book of Abstracts (Cluj-Napoca, Romania, July 2 2015), M. Frentiu, H. F. Pop, and S. Motogna, Eds., Babes-Bolyai University, (2015), 71-74.
[5] E. Achtert, S. Goldhofer, H. P. Kriegel, E. Schubert, A. Zimek: *Evaluation of Clusterings Metrics and Visual Support*, Proceedings of the 28th International Conference on Data Engineering (ICDE), Washington, DC, (2012), 12851288.
[6] B. S. Everitt, *Unresolved problems in cluster analysis*, Biometrics, (1979), 169-181.
[7] E. B. Fowlkes, C. L. Mallows, *A method for comparing two hierarchical clusterings*, Journal of the American statistical association, 78(383), (1983), 553-569.
[8] R. D. Găceanu, H. F. Pop, *An adaptive fuzzy agent clustering algorithm for search engines*. In: Macs 2010 8-Th Joint Conference on Mathematics and Computer Science. (2011), 185.
[9] C. Goutte, E. Gaussier, *A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation*, Xerox Research Centre Europe 6, (2004), 345-359.
[10] J. Han, M. Kamber, J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.
[11] P. Hansen, B. Jaumard, *Cluster analysis and mathematical programming*, Mathematical programming, 79(1-3), (1997), 191-215.
[12] J. A. Hartigan, *Clustering algorithms*. John Wiley Sons, (1975).
[13] D. B. Henry, P. H. Tolan, D. Gorman-Smith, *Cluster analysis in family psychology research*, Journal of Family Psychology, 19(1), (2005), 121.

[14] S. Kotsiantis, P. Panayiotis, *Recent advances in clustering: A brief survey*, WSEAS Transactions on Information Science and Applications, 1 (2004), 73-81.
[15] F. Kovcs, C. Legny, A. Babos, *Cluster validity measurement techniques*, In 6th International symposium of hungarian researchers on computational intelligence, (2005), 18-19.
[16] M. Lichman, *UCI Machine Learning Repository*, Irvine, University of California, School of Information and Computer Science, (2013).
[17] D. Lin, *An information-theoretic definition of similarity*, ICML Vol. 98, (1998), 296-304.
[18] J. Podani, *Multivariate exploratory analysis of ordinal data in ecology: pitfalls, problems and solutions*, Journal of Vegetation Science, 16, 5, (2005), 497-510.
[19] M. Halkidi, Y. Batistakis, M. Vazirgiannis, *On clustering validation techniques*, Journal of Intelligent Information Systems, 17(2), (2001), 107-145.
[20] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, *Understanding of internal clustering validation measures*, 10th International Conference on IEEE. (2010), 911-916.
[21] F. Murtagh, *A survey of recent advances in hierarchical clustering algorithms*, The Computer Journal, 26(4), (1983), 354-359.
[22] R. Real, *Tables of significant values of Jaccard's index of similarity*, Miscel lnia Zoolgica 22, no. 1 (1999): 29-40.
[23] E. Rendn, I. Abundez, A. Arizmendi, E. Quiroz, *Internal versus external cluster validation indexes*, International Journal of computers and communications, 5(1), (2011), 27-34.
[24] P. J. Rousseeuw, *Silhouettes: a graphical aid to the interpretation and validation of cluster analysis*, Journal of computational and applied mathematics, 20, (1987), 53-65.
[25] S. Salvador, P. Chan, *Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms*, In Tools with Artificial Intelligence, 2004, 16th IEEE International Conference on IEEE. (2004), 576-584
[26] R. Sibson, *SLINK: an optimally efficient algorithm for the single-link cluster method*, The Computer Journal, 16(1), (1973) 30-34.
[27] G. J. Szekely, M. L. Rizzo, *Hierarchical clustering via joint between-within distances: Extending Wards minimum variance method*, Journal of classification, 22(2), (2005), 151-183.
[28] R. Xu, D. Wunsch, *Survey of clustering algorithms*, Neural Networks, IEEE Transactions, 16(3), (2005), 645-678.

Department of Computer Science, Babeş-Bolyai University, 1 M. Kogălniceanu Street, 400084, Cluj-Napoca, Romania

E-mail address: {adrianac, rgaceanu, hfpop}@cs.ubbcluj.ro