Comparative Analysis of Various Clustering Algorithms Using WEKA

Priyanka Sharma

Asst. Professor, Dept. of Comp. Sci. and Apps., Chaudhary Devi Lal University, Sirsa, Haryana, India

---***---

Abstract - *Clustering is an unsupervised learning problem which is used to determine the intrinsic grouping in a set of unlabeled data. Grouping of objects is done on the principle of maximizing the intra-class similarity and minimizing the inter-class similarity in such a way that the objects in the same group/cluster share some similar properties/traits. There is a wide range of algorithms available for clustering. This paper presents a comparative analysis of various clustering algorithms. In experiments, the effectiveness of algorithms is evaluated by comparing the results on 6 datasets from the UCI and KEEL repository.*

Key Words: Clustering, WEKA, K-mean, Farthest First, Filterer, and CLOPE.

1. INTRODUCTION

Clustering algorithms are often useful in various fields like data mining, learning theory, pattern recognition to find clusters in a set of data. Clustering is an unsupervised learning technique used for grouping elements or data sets in such a way that elements in the same group are more similar (in some way or another) to each other than to those in other groups. These groups are known as clusters. Clustering[1] is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, marketing, libraries, insurance, world wide web and bioinformatics. Cluster analysis was originated in anthropology by Driver and Kroeber in 1932 and introduced to psychology by Zubin in 1938 and Robert Tryon in 1939[2][3]. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently cluster the elements. Generally used scheme used to find similarity between data elements are inter and intra- cluster distance among the cluster elements. We can show this with a simple example:

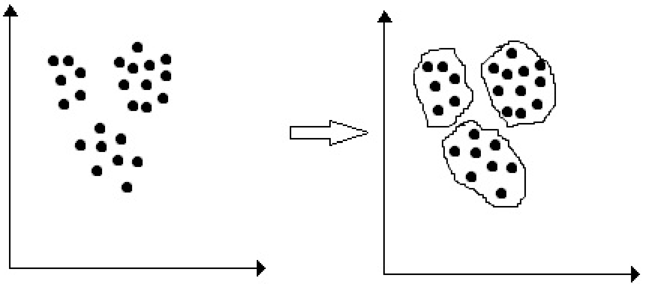

Fig -1: Clustering based on inter and intra distance measure.

Paragraph In the above example, data has been divided into three clusters using the similarity criterion "*distance*": two or more elements belong to the same cluster if they are "closer" according to a given distance. For optimizing the clusters, intra-cluster distance should be minimized and inter-cluster distance should be maximized. This clustering technique is called *distance-based clustering*.
Another kind of clustering is *conceptual clustering* in which two or more elements belong to the same cluster if they are conceptually same or similar. In other words, clusters are formed according to descriptive concepts, not according to distance measure, which is shown in the figure 2.

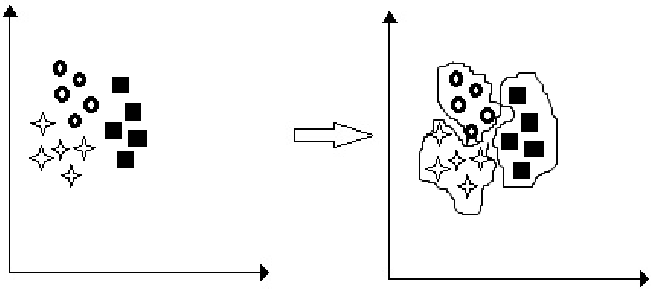

Fig -2: Clustering based on type of concepts of elements.

The appropriate clustering algorithm and parameter settings depend on the individual data set and intended use of the results. The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. Section 2 of paper presents clustering techniques to be compared. Section 3 gives an overview of WEKA. In section 4 and 5, experimental setup, performance measures and results have been shown. Section 6 concludes the paper.

2. CLUSTERING TECHNIQUES

A number of clustering techniques used in data mining tool WEKA have been presented in this section. These are:

2.1 CLOPE - Clustering with sLOPE [4]

Paragraph Like most partition-based clustering approaches, the best solution is approximated by iterative scanning of the database. However, criterion function is defined globally, only with easily computable metrics like size and width and is is very fast and scalable when clustering large transactional databases with high dimensions.

A transactional database *D* is a set of transactions $\{t_1, ..., t_n\}$. Each transaction is a set of items $\{i_1, ..., i_m\}$. A *clustering* $\{C_1, ... C_k\}$ is a partition of $\{t_1, ..., t_n\}$, that is, $C_1 \cup ... \cup C_k = \{t_1, ..., t_n\}$ and $C_i \neq \varphi$ and $C_i \cap C_j = \varphi$ for any $1 \leq i, j \leq k$. Each C_i is called a *cluster* and, *n*, *m*, *k* are used for the number of transactions, the number of items, and the number of clusters respectively.

Given a cluster *C*, all the distinct items (D(C)) in the cluster can be found with their respective *occurrences, Occ*(*i*, *C*) of item *i* in cluster *C*, that is, the number of transactions containing that item. Then the *histogram* of a cluster *C* can be drawn, with items as the *X*-axis, decreasingly ordered by their occurrences, and occurrence as the *Y*-axis for better visualization. *size S*(*C*) and *width W*(*C*) of a cluster *C* are defined below:

$$S(C) = \sum_{i \in D(C)} Occ(i, C) = \sum_{t_i \in C} |t_i|$$

$$W(C) = |D(C)|$$

The *height* of a cluster is defined as *H*(*C*)=*S*(*C*) / *W*(*C*).

It's straightforward that a larger height means a heavier overlap among the items in the cluster, and thus more similarity among the transactions in the cluster. To define the criterion function of a clustering, the shape of every cluster as well as the number of transactions in it has been taken into account. For a clustering C = $\{C_1, ..., C_k\}$, the following the criterion function has been used and found the most suitable.

$$Profit_r(\mathbf{C}) = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} \times |C_i|}{\sum_{i=1}^{k} |C_i|}$$

CLOPE is quite memory saving, even array representation of the occurrence data is practical for most transactional databases. The total memory required for item occurrences is approximately *M*×*K*×4 bytes using array of 4-byte integers, where *M* is the number of dimensions, and *K* the number of clusters. CLOPE is quite effective in finding interesting clusterings, even though it doesn't specify explicitly any inter-cluster dissimilarity metric.

2.2 Farthest First Clustering [11]

Farthest first [12] is a heuristic based method of clustering. It is a variant of K Means that also chooses centroids and assigns the objects in cluster but at the point furthermost from the existing cluster centre lying within the data area. Fast clustering is provided by this algorithm in most of the cases since less reassignment and adjustment is needed.

For each *Xi* = $[x_{i,1}, x_{i,2}, ..., x_{i,m}]$ in *D* that is described by *m* categorical attributes, $f(x_{i,j} \mid D)$ has been used to denote the frequency count of attribute value $x_{i,j}$ in the dataset. Then, a scoring function has been designed for evaluating each point, which is defined as:

$$Score(X_i) = \sum_{j=1}^{m} f(x_{i,j}|D)$$

In the farthest-point heuristic, the point with highest score is selected as the first point, and remaining points are selected in the same manner as that of basic farthest-point heuristic. Selecting the first point according to above defined scoring function could be fulfilled in *O* (*n*) time by deploying the following procedure (with two scans over the dataset):

(1). In the first scan over the dataset, *m* hash tables are constructed as basic data structures to store the information on attribute values and their frequencies where m is number of attributes.

(2). In the second scan over the dataset, with the use of hashing technique, in *O* (1) expected time, the frequency count of an attribute value in corresponding hash table can be determined.

Therefore, the data point with largest score could be detected in *O* (*n*) time. Time complexity of the basic algorithm is O (nk), where n is number of objects in the dataset and k is number of desired clusters. In basic FF clustering, first point is selected randomly. Farthest-point heuristic based method is suitable for large-scale data mining applications.

2.3 Filtered Clusterer [14]

In mathematics, a filter [14] is a special subset of a partially ordered set. For example, the power set of some set, partially ordered by set inclusion, is a filter. Let *X* be a topological space and *x* a point of *X*. A filter base *B* on *X* is said to cluster at *x* (or have *x* as a cluster point) if and only if each element of *B* has nonempty intersection with each neighborhood of *x*.

- If a filter base *B* clusters at *x* and is finer than a filter base *C*, then *C* clusters at *x* too.
- Every limit of a filter base is also a cluster point of the base.
- A filter base *B* that has *x* as a cluster point may not converge to *x*. But there is a finer filter base that does. For example the filter base of finite intersections of sets of the sub base B U N_x.

- For a filter base *B*, the set $\cap\{\mathrm{cl}(B_0) : B_0 \in B\}$ is the set of all cluster points of *B* (note: $\mathrm{cl}(B_0)$ is the closure of B_0). Assume that *X* is a complete lattice.
  - The limit inferior of *B* is the infimum of the set of all cluster points of *B*.
  - The limit superior of *B* is the supremum of the set of all cluster points of *B*.

B is a convergent filter base if and only if its limit inferior and limit superior agree; in this case, the value on which they agree is the limit of the filter base.

2.4 k-Mean Clustering

k-means clustering technique [24] is one of the simplest unsupervised learning techniquess that aim to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean value. Initially, k centroids need to be chosen in the beginning. The next step is to take instances or points belonging to a data set and associate them to the nearest centers. After finding k new centroids, a new binding has to be done between the same data set points and the nearest new center. Process is repeated until no more changes are done. Finally, this algorithm aims at minimizing intra cluster distance (cost function also known as squared error function), automatically inter cluster distance will be maximized.

$$Cost_{Fun} = \sum_{i=1}^{k} \sum_{p \in C_i} ||p - m_i||^2$$

where,

m_i – mean of i^th cluster,
C_i - i^th cluster and
p – point representing the object.

k-means clustering algorithm is fast, robust, relatively efficient and easier to understand. Time complexity of the algorithm is O(tknd), where n is number of objects/ points in the data set, k is number of predefined clusters, d is number of attributes/ dimension of each object, and t is the number of iterations until optimal clusters are not obtained. As it is a heuristic algorithm, there is no guarantee that it will converge to the global optimum and may also provide the local optima as final result depending upon initial cluster centers. Noisy data and outliers are not handled.

3. WEKA

WEKA (Waikato Environment for Knowledge Analysis) [25][26] is an open source, platform independent and easy to use data mining tool issued under GNU General Public License. It comes with Graphical User Interface (GUI) and contains collection of data preprocessing and modeling techniques. Tools for data pre-processing, classification, regression, clustering, association rules and visualization as well as suited for new machine learning schemes are provided in the package. It is portable since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform.

User interfaces

Weka's main user interface is the Explorer, but essentially the same functionality can be accessed through the component-based Knowledge Flow as well as the command line interface (CLI). There is also the Experimenter, which allows the systematic comparison of the predictive performance of Weka's machine learning algorithms on a collection of datasets.

The *Explorer* interface features several panels providing access to the main components of the workbench:

- The *Preprocess* panel has facilities for importing data from a database, a csv or an arff file, etc., and for preprocessing this data using a so-called filtering algorithm. These filters can be used to transform the data from numeric to discrete, to remove missing instances, to appropriately choose missing values and converting csv file to arff and vice versa.
- The Classify panel enables the user to apply classification and regression algorithms to the resulting dataset, to estimate the accuracy of the resulting predictive model, and to visualize errors. There are various type of classification algorithms like rule based, decision tree, naïve Bayesian, lazy, mi, misc etc. This paper make use of decision tree classification algorithms.
- The Associate panel attempts to identify all important interrelationships between attributes in the data with the help of association learners like apriori, filtered associator, predictive apriori etc.
- The *Cluster* panel gives access to the clustering techniques in Weka, e.g., the simple k-means, cobweb, DBSCAN, CLOPE algorithm to provide different kind of clustering's for different situations and usage of their results.
- The *Select attributes* panel provides algorithms for identifying the most predictive attributes in a dataset.
- The *Visualize* panel shows a scatter plot matrix, where individual scatter plots can be selected and enlarged, and analyzed further using various selection operators.

Extension packages

In version 3.7.2 of weka, a package manager was added to allow the easier installation of extension packages. Much functionality has come in weka through continuous extension and updates to make it more sophisticated.

4. METHODOLOGY & PERFORMANCE MEASURES

Clustering techniques discussed in section 3 have been compared with the help of WEKA. Steps followed in the analysis are:

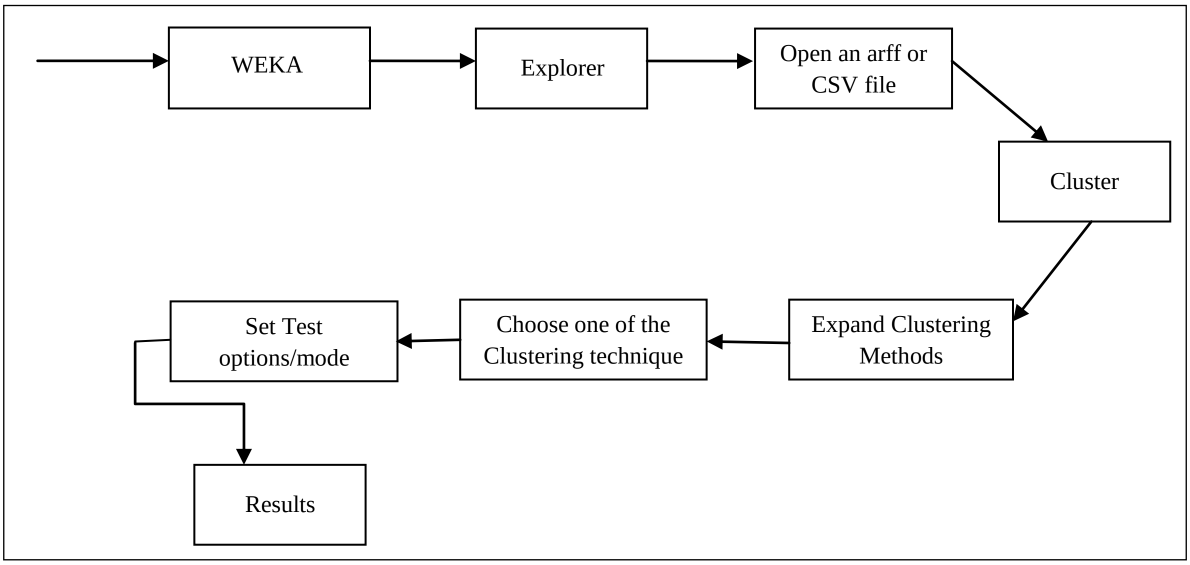

Fig -3: Clustering process used in WEKA.

As shown in Fig3, data file supplied should be in arff or CSV form. Data File should not contain unique id attribute like names, roll nos., remove these attribute either before supplying it for classification or untick these attribute before classification in WEKA. Note that Weka also provides access to SQL databases using Java Database Connectivity and can process the result returned by a database query. It is not capable of multi-relational data mining, but there is separate software for converting a collection of linked database tables into a single table that is suitable for processing using Weka

Performance measure used to determine accuracy of clustered data is *class to cluster evaluation*. A little about some important terms which are used in this measures is presented. These are:-

- True Clusterer (TC) – total number of elements belonging to clusters that were correctly predicted. These elements are verified using their classes i.e. TC= $TC_1 + TC_2 + \ldots\ TC_n$. Here n is the number of classes in the dataset and TC_i is the number of elements of class C_i which belongs to correct/right cluster.
- N – Total number of instances which are clustered.

Accuracy: It determines the proportion of the total number of instances clustered to the instances which are correctly clustered.

$$Accuracy = \frac{TC}{N}$$

5. EXPERIMENTAL RESULTS

A comparative analysis of various clustering algorithms has been made using six datasets taken from the KEEL [27] (a software tool to assess evolutionary algorithms in data mining problems) and UCI [28]machine learning repository. All the datasets are summarized in Table I.

Table -1: Datasets used in Experiments.

Datasets	#Instances	#Attributes	#Classes
Mushroom	5644	23	2
Car	1728	7	4
Iris	150	5	3
Tic-Tac-Toe	958	10	2
Breast Cancer	277	10	2
Chess	3196	37	2

Results are observed using two measures; accuracy and time, explained in section 4 using all the datasets mentioned in Table 1. Results have been shown in the Table 2, 3, 4, 5, 6, and 7.

Table -2: Comparison of Various Clustering Algorithms for Mushroom Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	45.92	2.14
Farthest First	66.16	0.07
Filtered Clusterer	51.44	0.39
k-Mean	51.44	0.32

Table -3: Comparison of Various Clustering Algorithms for Car Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	2.45	2.14
Farthest First	46.59	0.02
Filtered Clusterer	67.19	0.03
k-Mean	67.19	0.02

Table -4: Comparison of Various Clustering Algorithms for Iris Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	58.67	0.02
Farthest First	66	0
Filtered Clusterer	64.67	0
k-Mean	64.67	0

Table -5: Comparison of Various Clustering Algorithms for Tic-Tac-Toe Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	5.54	1.03
Farthest First	55.74	0.05
Filtered Clusterer	50.52	0.17
k-Mean	50.52	0.09

Table -6: Comparison of Various Clustering Algorithms for Cancer Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	10.84	0.27
Farthest First	74.27	0
Filtered Clusterer	55.96	0.06
k-Mean	55.94	0.03

Table -7: Comparison of Various Clustering Algorithms for Chess Dataset.

Clustering Method	Accuracy (%)	Time Taken (in secs.)
CLOPE	37.71	2.89
Farthest First	53.79	0.06
Filtered Clusterer	53.33	0.27
k-Mean	55.33	0.2

In the analysis, two different measures have been used for comparing various clustering algorithms. From the results obtained in the Tables 2, 3, 4, 5, 6, and 7, it can be seen that Farthest First performs best among all in most of cases. Clustering accuracy in Farthest First is maximum and time taken in clustering is minimum. CLOPE clustering has proven worst in all the cases. Its clustering accuracy is minimum as well as time taken is maximum. Rest of the models lies in between the best and worst ones.

6. CONCLUSIONS

Comparative analysis of various clustering algorithms has been made. The results have been validated using six datasets taken from UCI and KEEL repository and noticed that datasets are successfully clustered with a quite good accuracy. Few of the clustering techniques have better accuracy, others take less time, and many others have a trade-off between accuracy and time taken. Appropriate methods can be used according to their usage.

REFERENCES

[1] M. Mor, P. Gupta, and P. Sharma, “A Genetic Algorithm Approach for Clustering.”

[2] K. Bailey, “Numerical taxonomy and cluster analysis,” *Typol. Taxon.*, vol. 34, p. 24, 1994.

[3] R. C. Tryon, *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Edwards brother, Incorporated, lithoprinters and publishers, 1939.

[4] Y. Yang, X. Guan, and J. You, “CLOPE: a fast and effective clustering algorithm for transactional data,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 682–687.

[5] “Cobweb (clustering),” *Wikipedia, the free encyclopedia*. 06-Dec-2014.

[6] I. Jonyer, D. J. Cook, and L. B. Holder, “Graph-based hierarchical conceptual clustering,” *J. Mach. Learn. Res.*, vol. 2, pp. 19–43, 2002.

[7] "DBSCAN," *Wikipedia, the free encyclopedia*. 07-Feb-2015.

[8] M. Kryszkiewicz and P. Lasek, "TI-DBSCAN: Clustering with DBSCAN by Means of the Triangle Inequality," in *Rough Sets and Current Trends in Computing*, 2010, pp. 60–69.

[9] "Expectation–maximization algorithm," *Wikipedia, the free encyclopedia*. 18-Feb-2015.

[10] T. K. Moon, "The expectation-maximization algorithm," *Signal Process. Mag. IEEE*, vol. 13, no. 6, pp. 47–60, 1996.

[11] H. Zengyou, "Farthest-point heuristic based initialization methods for K-modes clustering," 2006.

[12] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Procs. of the twenty-first international conference on Machine learning*, 2004, p. 11.

[13] D. A. Vadeyar and H. K. Yogish, "Farthest First Clustering in Links Reorganization," *Int. J. Web Semantic Technol.*, vol. 5, no. 3, 2014.

[14] "Filter (mathematics)," *Wikipedia, the free encyclopedia*. 06-Dec-2014.

[15] "Hierarchical clustering," *Wikipedia, the free encyclopedia*. 09-Feb-2015.

[16] I. Davidson and S. S. Ravi, "Agglomerative hierarchical clustering with constraints: Theoretical and empirical results," in *Knowledge Discovery in Databases: PKDD 2005*, Springer, 2005, pp. 59–70.

[17] "Dendrogram," *Wikipedia, the free encyclopedia*. 08-Dec-2014.

[18] K. C. Gowda and T. V. Ravi, "Divisive clustering of symbolic objects using the concepts of both similarity and dissimilarity," *Pattern Recognit.*, vol. 28, no. 8, pp. 1277–1282, 1995.

[19] A. W. Edwards and L. L. Cavalli-Sforza, "A method for cluster analysis," *Biometrics*, pp. 362–375, 1965.

[20] K. C. Gowda and G. Krishna, "Agglomerative clustering using the concept of mutual nearest neighbourhood," *Pattern Recognit.*, vol. 10, no. 2, pp. 105–112, 1978.

[21] X. Wang and H. J. Hamilton, *DBRS: a density-based spatial clustering method with random sampling*. Springer, 2003.

[22] "OPTICS algorithm," *Wikipedia, the free encyclopedia*. 21-Dec-2014.

[23] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *ACM Sigmod Record*, 1999, vol. 28, pp. 49–60.

[24] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Morgan kaufmann, 2006.

[25] E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, "Weka," in *Data Mining and Knowledge Discov.*, Springer, 2005, pp. 1305–1314.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.

[27] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework.," *J. Mult.-Valued Log. Soft Comput.*, vol. 17, 2011.

[28] A. Asuncion and D. Newman, *UCI machine learning repository*. Irvine, 2007.

BIOGRAPHIES

Ms. Priyanka Sharma working as an assistant professor (cont.) in Chaudhary Devi Lal University, Sirsa, Haryana, India since Aug, 2014.

She has completed B.Tech (CSE)-2008-12, M.Tech (CSE)-2012-14 from Guru Jambheshwar University of Sci. & Tech., Hisar, Haryana, India.

She has qualified GATE and UGC-NET/JRF.

She is currently doing research in Data Mining and Knowledge Discovery.