Journal of Information Engineering and Applications
ISSN 2224-5782 (print) ISSN 2225-0506 (online)
Vol.5, No.4, 2015

Clustering Algorithm for Files and Data in Cloud (MMR Approach)

Shobhit Tiwari* Sourav Khandelwal
School of Computing Science and Engineering, Vellore Institute of Technology, Vellore, India

Abstract
The gradual advancement of technologies like cloud computing, there is enough necessity to improve computing techniques. Speed and security is a problem even in this environment. In order to achieve security, the files and data in any cloud environment need to be anonymized. Clustering has been used for long as a technique to achieve this. The cloud environment is supposed to have uncertainty and the data may be heterogeneous. So, a robust algorithm is necessary in this direction. In this paper we propose an algorithm, which we call as AMMR (Advanced Min-Min Roughness) algorithm by extending the MMR algorithm developed by Tripathy. This algorithm has the characteristics like being stable, handling uncertainty and categorical data.
Keywords : Cloud Computing, Database, MMR, Rough Set.

1. Introduction

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) . Data clustering is a method in which we make cluster of objects that are somehow similar in characteristics. The criterion for checking the similarity is implementation dependent. Clustering is often confused with classification, but there is some difference between the two. In classification the objects are assigned to predefined classes, whereas in clustering the classes are also to be defined. We can use clustering in different techniques in data mining. Unsupervised classification, data segmentation are some of its applications. We can easily segment the large heterogeneous data sets into smaller homogeneous subsets which are further analyzed after separately modeling. Biomedicine, research and application of radar scanning, development and manufacturing are some of the active areas where clustering techniques have been constantly being used successfully. Cloud computing is a potential area where optimization needs to be done with respect to data clustering. There are or may be not enough sufficient algorithm that can cluster the data . Moreover these algorithms are based more on numerical data , and are not designed to handle uncertainty in the clustering process. Dealing with the cloud where there are different types of files and data that have multi- valued attribute data and file clustering becomes a important issue.

There are algorithms related to application of fuzzy set in clustering categorical data have been proposed by Huang (Halkidi, Batistakis & Vazirgiannis ,2001) and Kim et al. (He, Xu & Deng, 2004). However, these algorithms require multiple runs to establish the stability needed to obtain a satisfactory value for one parameter used to control the membership fuzziness. So, a clustering algorithm that is robust and can, to an extent handle to an extent uncertainty in data clustering. This kind of algorithm was given by (Ganti, Gehrke & Ramakrishnan, 1999), which uses rough set theory. It manages impreciseness based on rough set theory. It also handles uncertainty to a great extent. Given a large data set it can provide with stable results given a input. In the following sections we present the new algorithm which can be considered the advanced version of MMR, analyze its superiority and using advanced clustering techniques cluster data sets in a cloud.

2. Cloud Computing

Cloud computing is a delivery platform which is flexible, cost-effective and proven. It basically provides business or consumer IT services over the Internet. The term Cloud is used as a metaphor for the Internet. Regardless of user location or device, Cloud resources can be rapidly deployed and easily scaled, with all processes, applications and services provisioned “on demand,”. It is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a metered service over a network. Most cloud computing infrastructures consist of services delivered through shared data-centers and appearing as a single point of access for consumers' computing needs. It can be viewed as a combination (have characteristics) of:

1. Client-server model: Distributed application that has a server and client used for communication between a server and client.
2. Autonomic computing: Self-management for Computer System.
3. Mainframe computer: for bulk data processing.
4. Peer to peer: in contrast to client server, both participants being at the same time both suppliers and consumers of resource without the need of central coordination.
5. Utility computing: Pay and use kind of tradition.

It is device and location independent. Its characteristic of multi tenancy allows sharing of resources and costs

across a large pool of users. It is reliable and scalable. Maintenance is also easier as cloud computing applications need not to be installed in each user's computer.

2.1 Types of cloud

1. Hybrid Cloud: It is a combination of two or more private, community or public cloud. It further provides opportunity for further deployment.
2. Private cloud: A cloud which is developed and deployed solely for a single organization. It can be either managed the organization itself or third party.
3. Intercloud : Extension of internet which is a interconnected global clouds of clouds.
4. Community cloud: It is a cloud that is shared between different organizations that can be either managed internally or by the third party.

There are three types of Cloud models:

1. SAAS: Software as a Service, which provides software as and when required.
2. PAAS: Platform as a Service provides platform, middleware, databases, development tooling etc when and as required.
3. IAAS: Infrastructure as a Service provides support of Servers, Networking, and Storage as and when required by the user.

3. MMR Algorithm

The MMR algorithm is used for clustering data tables with objects having multiple attributes. Here, this algorithm finds importance in PAAS (platform as a service) which provides database management and services. In general, in a cloud, we may have many objects with heterogeneous attributes say our data table may contain pictures, videos, audio etc. Now these objects may have different attributes for example a picture may have attribute of (date, size, format, location, dimensions).We may need to cluster the data tables according to our need for this we must first identify the pictures (say by format identification for e.g. file formats with .jpg, .jpeg, .gif are more likely to be pictures) and then extract the attributes in a data table, and cluster according to the algorithm discussed. These concepts can be further extended according to our needs by extracting the multiple attributes of the required files and then clustering the data tables.

4. Advanced MMR

Nomenclature used in Advanced MMR is shown in Table 1.

4.1 Algorithm

Select clustering type
Switch:
Case i) file type //*clustering based on type of file attribute;*
Loop{array[attributes]: = Getfile(attributes)};
Create a separate database of attributes collected through functions;
//*for example if (jpg, jpeg, png, gif file extensions etc are identified and placed under a column* named pictures)
//*Similarly for videos etc;*
Call procedure AMMR(U,k) //*clustering algorithm*
Break;
Case ii)data type;
Loop{array[attributes]:=Getdata(attributes)};
Create a separate database of attributes collected through functions;
Call procedure Ammr(U,k) //*clustering algorithm*
Break;
Procedure AMMR(U, k) //*after getting data table of attributes;*
Begin
Set current number of cluster CNC = 1
Set NodeParent = U
Loop1:
If CNC < k and CNC $\neq$ 1 then
ParentNode = ProcParentNode (CNC)
End if
// *Clustering the nodeParent*
For each $a_i \in A$ ($i = 1$ to n , where n is the number of attributes in A)
Determine $[x_m]_{Ind(a_i)}$ (m = 1 to number of objects)

For each $a_j \in A(j = 1 \text{ to } n$, where n is the number of attributes in $A, j \neq i)$ Calculate $Rough_{aj}(a_i)$
Next

Min-Roughness (a_i) = Min ($Rough_{aj}$ (a_i))
Next
Set Min–Min-Roughness =Min (Min-Roughness (a_i)), i = 1,. . .,n
Determine splitting attribute a_i corresponding to the Min–Min-Roughness
Do binary split on the splitting attribute ai
CNC = the number of leaf nodes
Go to Loop 1
End
ProcParentNode (CNC)
Begin
Set i = 1
Do until i < CNC
Size (i) = Count (Set of Elements in Cluster i)
i = i + 1
Loop
Determine Max (Size (i))
Return (Set of Elements in cluster i) corresponding to Max (Size (i))
End

5. Handling of different types of data

This AMMR algorithm is designed to cluster the data and files available in a cloud. It first extracts the attributes and other information from the network and creates a separate database for example if (jpg, jpeg, png, gif file extensions etc are identified and placed under a column named pictures) Similarly for videos etc. Then it uses the MMR logic to cluster the data sets. For clustering of numerical data we have defined classes. Here we have taken minimum of 'n' of the number of equivalence classes of all attributes containing categorical data, and does not divide the number of objects then its ceiling or floor is selected, whichever leads to merging of less number of objects. So, we merge the elements which are nearest possible. This iterative step ends when the above condition is satisfied. The merged set is termed as an element and all its elements are included in same class. The AMMR algorithm emphasizes on finding the minimum of roughness values for every equivalence class of each attribute and choosing the attribute possessing the equivalence class with least minimum as splitting attribute rather than choosing the attribute which has the least roughness with respect to any attribute.

5.1 Clustering Process

After getting the attributes in the database and after splitting the objects into two parts, we find the average distance between every two tuples in each cluster. For finding the distance we shall use the method of Hamming distance, which gives the difference between two objects, taking the equality or otherwise of the respective attributes into consideration. If the respective attribute values are equal we add zero and add 1 otherwise. Comparing the average distances in the clusters, the record having the least distance is selected. If clusters with same average distances are found then we choose the one which has more number of objects. In case of a tie a random selection is made.

6. Conclusions

No or very few algorithms have described approach to cluster the data, files and network in a cloud. Here we have introduced a criterion to find a distance between any two data objects by generalizing Hamming distance. The object is created by selecting attributes from files and networks. Our Other achievement is in choosing cluster for re-clustering and handling of heterogeneous data and files. Future enhancements of this algorithm can be made in the fields of selection of splitting attribute by introducing fuzzy properties. This will lead to development of rough- fuzzy concepts in clustering.

References

Dempster, A., Laird, N., Rubin, D. (1977), Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1),pp 1–38.
Ganti, V., Gehrke, J. Ramakrishnan, R. (1999), CACTUS – clustering categorical data using summaries, in: Fifth

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 73–83.
Gibson, D., Kleinberg, J., Raghavan, P. (2000), Clustering categorical data: an approach based on dynamical systems, The Very Large Data Bases Journal 8 (3–4), pp 222–236.
Guha, S. Rastogi, R. Shim, K. (2000), ROCK: a robust clustering algorithm for categorical attributes, Information Systems 25 (5) , pp 345–366.
Halkidi, M., Batistakis, Y., Vazirgiannis, M. (2001), On clustering validation techniques, Journal of Intelligent Information Systems 17 (2–3) , pp 107–145.
Han, E., Karypis, G., Kumar, V., Mobasher, B. (1997), Clustering based on association rule hypergraphs, in: Workshop on Research Issues on Data Mining and Knowledge Discovery, pp. 9–13.
He, Z., Xu, X., Deng, S. (2004): A link clustering based approach for clustering categorical data, Proceedings of the WAIM Conference, <http://xxx.sf.nchc.org.tw/ftp/cs/papers/0412/0412019.pdf>.
He, Z., Xu, X., Deng, S. (2002), Squeezer: an efficient algorithm for clustering categorical data, Journal of Computer Science & Technology 17(5), pp 611–624.
Huang, Z. (1998), Extensions to the k-means algorithm for clustering large data sets with categorical values, Data Mining and Knowledge Discovery 2 (3), pp 283–304.
Kim, D., Lee, K., Lee, D. (2004), Fuzzy clustering of categorical data using fuzzy Centroid Pattern
Krishnapuram, R., Frigui, H., Nasraoui, O. (1995), Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation, IEEE Transactions on Fuzzy Systems 3 (1), pp 29–60.
Krishnapuram, R., Keller, J. (1993), A possibilistic approach to clustering, IEEE Transactions on Fuzzy Systems 1 (2), pp 98–110.
Pawlak, Z. (1991), Rough Sets – Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Boston.
Pawlak, Z. (1997), Rough set approach to knowledge-based decision support, European Journal of Operational Research 99 (1), pp 48–57.
Ralambondrainy, H. (1995), A conceptual version of the K-means algorithm, Pattern Recognition Letters 16 (11), pp 1147–1157.
Ruspini, E. (1969), A new approach to clustering, Information Control 15 (1), pp 22–32.
Zhang, Y., Fu, A., Cai, C., Heng, P. (2000), Clustering categorical data, in: Proceedings of the 16th International Conference on Data Engineering, pp. 305–324.

Sourav Khandelwal is currently pursuing his B.Tech in Computer Science and Engineering from Vellore Institute of Technology. His research interests include Analytical Problem solving, Cloud Computing. He also has a passion for coding.

Shobhit Tiwari is currently pursuing his B.Tech in Computer Science and Engineering from Vellore Institute of Technology. His research interests include Wireless Networks, E-learning and Algorithm Design.

Table 1. Nomenclature used in Advanced MMR Algorithm

Symbol	**Meaning**
U	Universe or the set of all objects (x1, x2, . .)
X	X subset of the set of all objects, (X ⊂ U)
x_i	object belonging to the subset of the set of all objects, xi ∈ X
A	the set of all attributes (features or variables)
a_i	attribute belonging to the set of all attributes, ai ∈ A.
V(a_i)	set of values of attribute ai (or called domain of ai)
B	non-empty subset of A(B ⊆ A)
X_B	lower approximation of X with respect to B
$\overline{X_B}$	upper approximation of X with respect to B
$R_{ai}(x)$:	roughness with respect to {ai}
$[X_i]_{Ind(B}$	Equivalence class of x_i in relation Ind(B), also known as elementary set in B.

The IISTE is a pioneer in the Open-Access hosting service and academic event management. The aim of the firm is Accelerating Global Knowledge Sharing.

More information about the firm can be found on the homepage:
http://www.iiste.org

CALL FOR JOURNAL PAPERS

There are more than 30 peer-reviewed academic journals hosted under the hosting platform.

Prospective authors of journals can find the submission instruction on the following page: http://www.iiste.org/journals/ All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Paper version of the journals is also available upon request of readers and authors.

MORE RESOURCES

Book publication information: http://www.iiste.org/book/

Academic conference: http://www.iiste.org/conference/upcoming-conferences-call-for-paper/

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar