Hybridization of Fuzzy Clustering and Hierarchical Method for Link Discovery

Enseih Davoodi Jam
Department of Computer Science, University of Isfahan, Isfahan, Iran
nc_davodi.softcom@yahoo.com

Mohammadali Nematbakhsh
Department of Computer Science, University of Isfahan, Isfahan, Iran
nematbakhsh@eng.ui.ac.ir

Mojgan Askarizade
Department of Computer Science,University of Isfahan, Isfahan, Iran
Mojgan.askarizade@gmail.com

Abstract

Clustering is an active research topic in data mining and different methods have been proposed in the literature. Most of these methods are based on numerical attributes. Recently, there have been several proposals to develop clustering methods that support mixed attributes. There are three basic groups of clustering methods: partitional methods, hierarchical methods and density-based methods. This paper proposes a hybrid clustering algorithm that combines the advantages of hierarchical clustering and fuzzy clustering techniques and considers mixed attributes. The proposed algorithms improve the fuzzy algorithm by making it less dependent on the initial parameters such as randomly chosen initial cluster centers, and it can determine the number of clusters based on the complexity of cluster structure. Our approach is organized in two phases: first, the division of data in two clusters; then the determination of the worst cluster and splitting. The number of clusters is unknown, but our algorithms can find this parameter based on the complexity of cluster structure. We demonstrate the effectiveness of the clustering approach by evaluating datasets of linked data. We applied the proposed algorithms on three different datasets. Experimental results the proposed algorithm is suitable for link discovery between datasets of linked data. Clustering can decrease the number of comparisons before link discovery.

Keywords: Hierarchical method, Fuzzy Clustering, similarity measure, Linked Data

1. Introduction

The Linked Data movement has experienced exponential growth in terms of published data sets. Within two years, the number of triples has grown from 4.7 to 34 billion and therefore there is a lack of Linked Discovery techniques to find more and more links between knowledge bases. The task of a link discovery is to compare entities and suggests a set of entities whose similarity is above a given threshold. Clustering can decrease the number of comparisons before link discovery. Clustering is an important tool for analyzing data. Clustering is the process of grouping a data set in such a way that the similarity between data within a cluster is maximized while the similarity between data of different clusters is minimized. A number of clustering techniques have been developed, and these can be classified as hierarchical, partitional and density-based methods. Hierarchical techniques produce a nested sequence of partitions, which a single, all inclusive cluster at the top and singleton clusters of individual points at the bottom. Agglomerative and divisive are two types of hierarchical clustering methods [1]. Agglomerative clustering methods start with each object in a distinct cluster and successively merge them to larger clusters until a stopping criterion is satisfied. Alternatively, Divisive hierarchical clustering started with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of cluster is obtained or the diameter of each cluster is within a certain threshold. From another perspective, clustering algorithms can be classified into two categories, hard clustering and fuzzy clustering. While in hard clustering an entity belongs only to one cluster but Fuzzy clustering methods, allow the entities to

belong to several clusters simultaneously, with certain degrees of membership. The memberships help us discover more advanced relations between a given entities and the disclosed clusters [2].
Related works that focus on Linked Data include Bizer et al[3] who presented a "Multidimensional Blocking" for link discovery. This method is organized in three phases:

- Index generation
- Index aggregation
- Comparison pair generation

The overhead of" Multidimensional Blocking" is higher than that of standard blocking. The current known framework for link discovery on the Web is SILK. It provides a flexible, declarative language for specifying link condition. The weakness of SILK is that the recall is not guaranteed to occur [4]. In this paper, we present a hybrid clustering algorithm that combines the advantages of hierarchical clustering and fuzzy clustering techniques. Our algorithm cluster similar entities of data sets and reduce the number of comparisons before link discovery. Our approach is organized in two phases:

1. First based on feature selection principles, the properties of entities are selected. Then the entities are divided into two clusters by random initialization of cluster centers.
2. In the split phase, the worst cluster is determined and split. This stage is repeated until the optimal number of clusters is achieved.

The most important advantage of our approach is:

- The intelligent finding of the number of clusters.
- The ability to run on metric and semi- metric space.
- The consideration of the various types of entity properties.
- Less dependent on randomly chosen initial cluster centers.

The remainder of this paper is structured as follow: a formal definition for the problem is presented in section 2. We present the approach in Section 3 and report on the results of experimental evaluation in Section 4. We conclude with the discussion and an outlook on future work in Section 5.

2. Problem formulation

Given two relations with the same features *RA (f1, f2 …, ft)* and *RB (f1, f2 …, ft)*. A fuzzy matching function *FMF* takes as input triple ($r_A, r_B, \{\theta_1, \dots \theta_t\}$) and produces a fuzzy output {[0,1]} where:

- *rA* ∈ *RA* is an entity with attribute values *(rA(f1),…,rA(ft))* and *rA(f1)* ∈ *Dom(RA.f1), …, rA(ft)* ∈ *Dom(RA.ft)*.
- *rB* ∈ *RB* is a record with attribute values *(rB(f1),…,rB(ft))* and *rB(f1)* ∈*Dom(RB.f1),…,rB(ft)* ∈ *Dom(RB.ft)*.
- $\{\theta_1, \dots \theta_t\}$ are predefined similarity thresholds for the corresponding attributes *f1, … ft* in *RA* and *RB*.

The output of the fuzzy matching function *FMF* is decided based on:

$$FMF(r_A, r_B, \{\theta_1, \dots \theta_t\}) = \left\{ fuzzy\,num = avarage\left(g_i(r_A(f_i), r_B(f_i))\right)\right\} \quad (1)$$

Where $g_i: Dom(R_A.f_i) * Dom(R_B.f_i) \rightarrow R^+, i = 1, \dots t$ are predefined similarity measures or distance functions defined over the domains of corresponding attribute *fi* for the relations *RA* and *RB*.

3. Approach

In this section, we present our model in more detail. Figure 1 gives an overview of the workflow. Our approach is organized in two phases: first, the division of data in two clusters; then

the determination of the worst cluster and splitting. The number of clusters is unknown, but our algorithms can find this parameter based on the complexity of cluster structure.

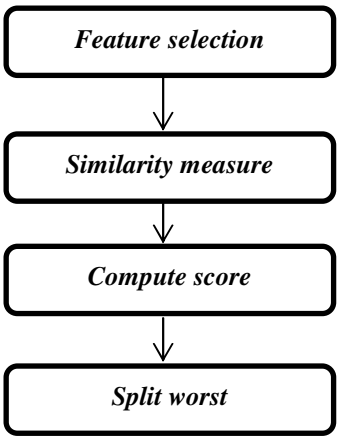

Figure 1. Workflow of approach

Initialization of clustering

Our approach is a branch of divisive hierarchical clustering. Divisive hierarchical clustering started with all objects in one cluster. It subdivides the cluster into smaller pieces, until each object forms a cluster on its own.

Feature selection

Clustering activity is based on feature selection. Feature selection is the process of identifying the most effective subset of the original feature to use in clustering. The best subset contains the least number of dimensions. Subset selection algorithms can be classified into Wrappers, Filters. A wrapper method evaluates the candidate feature subsets by the learning algorithm. In clustering, a wrapper method uses a clustering algorithm. Filter methods use an evaluation function that bases on properties of the data and is independent on any algorithm. Advantages of filter method are that they scale to high-dimensional datasets and they are computationally simple and fast but the wrapper methods have higher accuracy [5]. We use Wrapper method and evaluate the subset with an enhancement of fuzzy clustering.

Similarity measure

It is natural to ask what type of standards we should use to determine the closeness, or how to measure the distance or similarity between a pair of entities. Extracting features from entities is based on the domain of the datasets; a similarity measure is assigned to the features. Since datasets of linked data typically involve a variety of different data types, various similarity measures are defined. Most of the well-known clustering methods are implemented only for numerical data. The proposed clustering algorithm allows different types of data features such as numeric, symbolic and string data.

Numeric similarity:

The similarity of two numbers is computed with:

$$sim_{number}(x,y) = \begin{cases} 1 - \left(\frac{|x-y|}{max(x,y)}\right) & if\ x,y > 0\ \ or\ \ x,y < 0 \\ -\left(\frac{|x-y|}{max(x,y)}\right) & if\ x > 0, y < 0\ \ or\ \ x < 0, y > 0 \end{cases} \quad (2)$$

String similarity:

A number of string similarity measures have been developed and presented in the literature. We use the "Jaro-winkler"[6]:

$$Jaro(\sigma_1,\sigma_2) = \frac{1}{3}\left(\frac{c}{|\sigma_1|} + \frac{c}{|\sigma_2|} + \frac{c-t/2}{c}\right) \quad (3)$$

Consequently, the following function is defined:

$$\mathbf{sim_{string}(x,y) = jaroalgoritm(x,y)} \quad (4)$$

Final similarity:

This function is the weighted average of the above similarity measures:

$$\mathbf{finalsim(a,b) = \frac{(sim_{number} * w_{num}) + (sim_{string} * w_{string})}{w_{total}}} \quad (5)$$

Initial cluster centers

The proposed method chooses two randomly- selected entities as the initial centers. After the initial cluster centers have been selected, each entity is assigned to the closest cluster, based on its distance from the cluster centers. For each clustering step, calculates membership matrix based on fuzzy clustering algorithm:

$$U_{i,j}^{t+1} = 1 \Big/ \sum_{l=1}^{c} \left(\frac{D_{l\,j}}{D_{i\,j}}\right)^{\frac{2}{2-m}} \quad (6)$$

For i=1,2,…..,C and j=1,2,……,N
C is the number of cluster
N is the number of entities.
D is distance the two entities.

Fuzzy clustering methods, allow the entities to belong to several clusters simultaneously, with certain degrees of membership. The memberships help us discover more advanced relations between a given entities and the disclosed clusters [2].After this step, all entities are divided into two clusters. To continue in the next step, we need to find new centers. For this purpose the following steps are carried out:

a. To find the entities which have the highest similarity to their centers, the cluster members are sorted in descending order.
b. 20 percent of previously sorted entities are listed under t as "sim list".An average of the feature of entities of "sim list" is calculated. For numeric properties, the average number and for string properties, the LCS algorithm[18] is used.
c. Finally, a new cluster center is computed with:

$$center = \frac{(average(number) * w_{num})}{w_{total}} + \frac{(LCA(string) * w_{string})}{w_{total}} \quad (7)$$

The pseudo-code of the proposed algorithm is represented in Algorithm 1.

[1] Longest Common Subsequence

Input: cluster c with its entities
Output: new center.
Begin
Entities are sorted in descending order.
Simlist ← %20 sorted entities.
Wn ←weigh of numeric properties.
Wstr ← weigh of string properties.
Wsym ← weigh of symbolic properties
Calculate average;
Begin
foreach *entities € simlist*
begin
anp ← average number properties (a,b)=a+b/2
astrp ← average string properties(a,b)=LCS(a,b)
end
end
*newcnetr=((anp*wn)+(astrp*wstr)/ total weight*
end.

Split cluster

The general idea in the splitting is to identify the "worst" cluster and split it, thus increasing the number of clusters one by one [7]. To find the worst cluster, point(i) is assigned to each cluster *i*:

$$point(i) = \frac{\sum_{k=1}^{n} U_{k\,i}}{n} \quad (8)$$

Small point (i) shows that cluster i is large and spare in distribution. Hence, the cluster which takes of minimum of point (i) will be the candidate for worst cluster.
Cluster **W** is identified to be split, supposing that the cluster center is C_0 and the number of clusters for each step is **C**. the algorithms for splitting can be formulated follows:

1. From among the entities of **W** the one labeled "not try" and has the lowest similarity with the **C-1** cluster centers is chosen and namedC_1.
2. The distance of all entities of **W** from C_0 andC_1 are calculated and the **W** cluster is split into W_0 and W_1 on basis of the calculated distance.
3. Calculate distance of each entities from C_0 andC_1. Then split cluster **W** into W_0 and W_1 based on calculated distance. C_1 Is assigned as the c[th] cluster center if $|W_1|/|W| > \%20(W)$ otherwise the label of C_1 should be changed to "try" and go to step 2.

After two steps, a new cluster is created. Step 1 and 2 are repeated with the reminding entities of **W** until C+1[th] clusters are found. The pseudo-code of the proposed algorithm is represented in Algorithm 2.

Input: cluster ***W***, centerC_0,
Output: new cluster W_0 and W_1
Begin
Find C_1
Begin
Foreach *entities* ∈ *W*
$C_0 \rightarrow$ *Entity has minimum distance with c-1 center and not tested.*
End.

Spilt w
Begin
Foreach *entities* ∈ *W*
Begin
dis1→ Calculate distance (e_i, C_0)
dis2→ Calculate distance (e_i, C_1)
If *(dis1>dis2)*
$W_0 : \{e_1\}$
Else
$W_1 : \{e_1\}$
End.
If $|W_1|/|W| > \%20(W)$
Split W to W_0 and W_1
Else e_i *is tested.*
End.
End.

The split worst cluster phase should be repeated until the optimal number of clusters is achieved. Choosing this parameter is a difficult problem. In each step of the algorithm, if the number of new cluster member is more than 20 percent of candidate cluster members, the split occurs. Based on the distribution, clusters broken or stops and the optimal number of clusters are obtained.

4. Evaluation

Datasets description

To prove the efficacy of the proposed approach, the performance of the clustering algorithm has been tested on three datasets of linked data: DBpedia, LinkedGeoData[19] and LinkedMDB datasets. DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web that currently contains more than 3.64 million resources[8]. Another dataset is LinkedGeodata that consist of 20 billion triples and the LinkedMDB database contains 3.5 million of RDF triples . We used a dataset consisting of 100,000 triples from DBpedia, 200,000 triples from LinkedGeoData for experiment 1 and 1000 triples from LinkedMDB and 1000 triple from DBPedia for experiment 2.

Performance Evaluation

Experiment 1: First, we interlinked places of DBpedia and LinkedGeoData datasets without the use of any clustering method. The result was $2 * 10^{10}$comparisons. Then, we evaluated how the clustering method reduces the number of comparisons. Table 1 summarizes the results. The evaluation shows that clustering reduces the number of comparisons by a factor of 142,857.

Table 1. Result of experiment1

Method	Comparisons
Full evaluation	$2 * 10^{10}$
Clustering	140,000

[2] http://www.linkedgeodata.org

Figure 2. Decrease comparisons.

Experiment 2: The proposed model does not depend on any specific domain, so we evalute our model with the data sets on a different domain. In this experimnet, the movies of LinkedMDB and DBpedia data set are linked. Table 2 summarizes the results.

Table 2. Results of experiment 2

Method	Comparisons
Full evaluation	1000000
Clustering	900

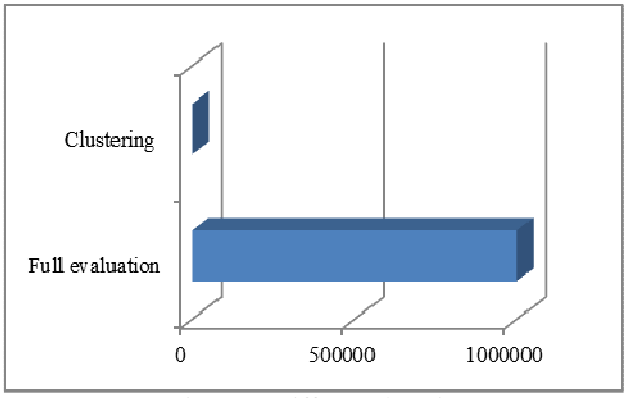

Figure 3. Different domain

Quality Evaluation

This section provides a summary of the quality evaluation of the implemented model. After clustering, each entities of datasets falls into one of the following groups:

- The entities which were recognized in same cluster and this recognition are correct.
- The entities which were recognized in same cluster but this recognition are incorrect.
- The entities which were recognized in different clusters but this recognition are incorrect and they are same in real.

In order to evaluate the quality of the interlinking LinkedGeoData and DBpedia, 500 place of LinkedGeoData which currently have correct owl:sameAs link to corresponded places in DBpedia are randomly selected. The results show that how many entities fall within each of the above defined groups.

Table 3. Interlinking between LinkedMDB and DBpedia

Type of group	count
Correct derived entities	475
Incorrect derived entities	25
Not- derived entities	25

The two most frequent and basic measures for information retrieval effectiveness are precision and recall.

Precision (P) is the fraction total of number detection similar entities that are true:

(9)

$$Precision = \frac{true\ detection\ entities}{detection\ entities} \quad (10)$$

Recall (R) is the fraction total of number similar entities that are true detection:

$$Rceall = \frac{true\ detection\ entities}{total\ entities} \quad (11)$$

Results from Table 4 and equations 9, 10 shows that the precision is %100 and recall is %95.
Our algorithm is an efficient clustering algorithm has some features that are mentioned in Table 4.

Table 4. Feature of our algorithm

Features	Y/N
Scalability	Y
Ability to cluster different types of attributes	Y
Ability to discover clusters with different shapes	N
Minimal input parameter	Y
Not sensitive to noise	N
Insensitive to the order of input records	Y
Ability to handle high dimensionality	Y

We explained the following two central evaluation questions:

- What is the best number of cluster?

Different performances of the tests indicate the number of clusters depend on the data sets and dispersal of their members is between $\sqrt{N}/_2$ and $\sqrt{N}$.

- Does the initial selection of centers affect the result?

The proposed algorithm was run with different initial centers and results show a random selection of centers does not affect the final results.

5. Discussion and Future Work

In this paper, we propose a new hybrid algorithm, which combines the features of fuzzy algorithm and hierarchical algorithm. Our algorithm decreases the number of comparisons on link discovery. Using hierarchical algorithm in the first level, the data is divided into two groups. In the second level the worst cluster is determined by matrix memberships and then it split. This stage is repeated until the optimal number of clusters is achieved.Creating typed links, between the entities of different datasets is one of the key challenges on web of data .We presented the clustering approach, which decreases the number of comparisons on link discovery. The results of linking the movies in LinkedMDB to corresponding movies in DBpedia and also linking the places in LinkedGeoData to the places of DBpedia show the it reduces the number of comparisons without loss of recall and precision. Hopefully in the future, we will be able to elevate the proposed method recall to 100 % using the membership matrix.

References

[1] Rui Xu , Donald C. Wunsch II, Survey Of Clustering Algorithms, IEEE Trans.on Neural Networks, Vol.16, No.3, May 2005, pp.645-678.
[2] Frank Höppner, Frank Klawonn, Rudlof Kruse, Fuzzy Cluster Analysis, Methods for Classification, Data Analysis, and Image Recognition. New York:Wiley, 1999.
[3] R. Isele, A. Jentzsch, C. Bizer, "Efficient Multidimensional Blocking for Link Discovery without losing Recall," Fourteenth International Workshop on theWeb and Databases (WebDB 2011), June 12, 2011 - Athens, Greece.
[4] J. Volz, C. Bizer, M. Geaedlke, G.Kobilarov, "Silk-A Link Discovery Framework for the web of data," LDOW 2009, April 20, 2009, Madrid, Spain.
[5] Anli K. Jain, M Narasimha Murty, Patrick J. Flynn, Data Clustering: A Review, ACM Computing Surveys, Vol. 31, No. 3, September 1999.
[6] Gonzalo Navarro, A guided tour to approximate string matching, ACM Comput. Surv., 33, 2001.
[7] Haojun Sun, Shengrui Wang, Qingshan Jiang, FCM-Based Model Selection Algorithms for Determining the Number of Cluster, Pattern Recognit 37:2027-2037.
[8] Christopher Sahnwldt, (2011,Julay). DataSet. [Online]. Available: http://wiki.dbpedia.org/Datasets.
[9] Christian. Bizer, Tom. Heath, Tim. Berners-Lee, Linked Data - The Story So Far, Journal on Semantic Web and Information Systems, Special Issue on Linked Data, 2009.
[10] Okite Hassanzadeh, Reynold Xin, Renee J.Miller, etal, Linkage query writer, Proceedings of the VLDBE ndowment, 2(2):1590–1593, 2009.
[11] Janson Tong-Li Wang, Xiong Wang, King-Ip Lin, et al, Evaluating a class of distance-mapping algorithms for data mining and clustering, In proceeeding of the 5th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 1999.data. ACM, 1995..
[12] Volker Roth, Julian Laub, Motoaki Kawanabe, Joachim M. Buhmann, Optimal cluster preserving embedding of nonmetric proximity data, IEEE Transactions on Pattern Analysis and Machine Intelligence (2003).
[13] Julius Volz, Christian Bizer, Martin Geaedlke, G.Kobilarov, Silk-A Link Discovery Framework for the web of data, LDOW 2009, April 20, 2009, Madrid, Spain.
[14] Miin-Shen Yang, Pei-Yuan Hwang, De-Hua Chen, Fuzzy clustering algorithms for mixed feature variables, Fuzzy Sets and Systems 141, pp. 301–317, 2004.