Clustering Users' Requirements Schemas

Nouha Arfaoui and Jalel Akaichi
BESTMOD – Institut Supérieur de Gestion, 41, Avenue de la liberté,
Cité Bouchoucha, Le Bardo 2000, Tunisie

Keywords: Star Schema, User Requirement, Clustering, k-Mode Extension, Ontology.

Abstract: Data Mining proposes different techniques to deal with data. In our work, we suggest the use of clustering technique since we want grouping the schemas into clusters according to their similarity. This technique is applied to variety type of variables. We focus on categorical data. Many algorithms are proposed, but no one of them takes into consideration the semantic aspect. For this reason, and in order to ensure a good clustering of the schemas of the users' requirements, we extend the k-mode algorithm by modifying its dissimilarity measure. The schemas within each cluster will be merged to construct the schemas of the data mart.

1 INTRODUCTION

Clustering is the unsupervised classification of patterns into groups called Clusters (Jain et al., 1999). It involves dividing a set of data points into non-overlapping groups, or cluster of points (Faber, 1994). The purpose of the cluster is to maximize the homogeneity of a partition of a set of variables into K disjoint clusters (Chavent et al., 2010). There are two different ways to classify the cluster analysis methods (Rezankova, 2009): the partitioning methods which cluster the objects into certain number of clusters and methods of hierarchical cluster analysis which make assignment of objects into different numbers of clusters possible.

The clustering can be applied to different kind of data: numeric data, binary data, categorical data and ordinarily data.

In this work, we look for clustering the schemas corresponding to the users' requirements so that in one cluster we will have the set of schemas which are semantically very close to be able to build the schemas of the data mart later. The existing algorithms do not take into consideration the semantic aspect while comparing the schemas. For this reason, we propose a new algorithm which is an extension of k-mode algorithm. This choice will be argue in the state of the art.

We modify the dissimilarity measure and we integrate the ontology to deal with the semantic aspect. Let us give a short presentation of the ontology. Indeed, it is used, at the beginning, with the philosophy. It is concerned with the nature of existence and the cataloguing of the existing entities (Quine, 1980). It is considered also as a collection of abstract objects, relationships and transformations that represent the physical and cognitive entities necessary for accomplishing some task (Alexander et al., 1986). It is used mainly to resolve the heterogeneity problem existing in the information environments (Alexiev et al., 2005). In our case, it is used to improve the document quality using the hierarchical knowledge (Hotho et al., 2003), (Jing et al., 2006).

The proposed algorithm can be applied in several areas. We propose its use to build the data mart schemas from the users' requirements that are presented as star schemas. The different users have different skills and belong to different departments, which makes the clustering of the schemas a crucial step to facilitate the schema building.

The outline of this work is as following:

- In the second section, we present the state of the art where we describe some of the existing clustering algorithms used to cluster the categorical data. We also give a comparative study to argue the extension of k-mode.
- In the third section, we focus on the collection of the users' requirements and we give the structure of the generated schemas.
- In the fourth section, we give the characteristics of the k-mode algorithm.

Arfaoui N. and Akaichi J..
Clustering Users' Requirements Schemas.
DOI: 10.5220/0004991100150021
In *Proceedings of 3rd International Conference on Data Management Technologies and Applications* (DATA-2014), pages 15-21
ISBN: 978-989-758-035-2

- In the fifth section, we describe our proposed algorithm. We give its steps, its new dissimilarity measure, as well as, the ontology that serves to improve the quality of the comparison.
- In the sixth section, we present the implementation of our solution.
- We finish this work with the conclusion and future work.

2 STATE OF THE ART

In this section, we summarize some existing algorithms used to cluster the categorical data. Then we give a comparative study in function of their complexity to argue the extension of k-mode.

2.1 The Clustering Algorithms

K-Mode: The k-means has the capacity to deal with large databases (San et al., 2004) and it is efficient with numerical data (Huang, 2008), (Ng et al., 2007), but it does not work with categorical data. The idea is to extend this algorithm to deal with real world data (including categorical data), hence the appearance of a new algorithm k-mode (Huang, 2008), (Ng et al., 2007), (San et al., 2004).

ROCK and QROCK: In (Guha et al., 2000), the authors present the ROCK (**RO**bust hierarchical **C**lustering with lin**K**s) which is a hierarchical clustering algorithm. Since the distance is not appropriate to deal with categorical data, they propose the links to measure the similarity/proximity between a pair of data points.

An improvement has been proposed in (Khan and Kant, 2007) through QROCK (Quick ROCK) that computes the clusters by determining the connected components of the graph which ensures having a drastic reduction of the computing time compared to ROCK.

COOLCAT: The proposed algorithm in (Barbara et al., 2002) uses the notion of entropy measure to group the records. The choice of the entropy is because it is a more natural and intuitive way of relating records and it does not rely in arbitrary distance metrics.

LIMBO: According to (Andritsos et al., 2004), it is built on the Information Bottleneck (IB) framework that is used to define a distance measure for categorical tuples. It has the capacity to produce clustering of different sizes in a single execution. Indeed, there is no need to keep whole tuples or clusters in the memory, but instead it is sufficient statistics to describe them.

MULIC: According to (Andreopoulos et al., 2004), MULIC is an extension of k-mode algorithm. It starts by clustering objects with high values; it does not require specifying the number of the clusters in the beginning since it can change during the process, it forms clusters gradually and if a new pattern is discovered, etc.

HIERDENC: The authors (Andreopoulos et al., 2004) applied this algorithm to categorical datasets. It clusters the m-dimensional cube with m-categorical attributes. The algorithm considers an object's neighbors that are within a radius of maximum dissimilarity in order to find the dense subspace. It starts from the densest subspace of the cube and it expands outwards from a dense subspace by connecting nearby dense subspaces.

RAHCA: In (Chen et al., 2006), the authors propose the use of Rough Set Theory (RST) to cluster categorical data in order to solve the problem of similarity measure. The clustering data set is mapped as the decision table through the introduction of decision attribute. RST is based on Euclidean distance.

2.2 Comparative Study

In order to justify our choice, we compare the previous algorithms according to their complexity as present in Table 1.

According to Table 1, we can notice that k-mode has the lowest complexity (O (n)), also, it has the capacity to deal with huge amount of data and it is easy to implement. For the cited reasons, we use it. K-mode, as it is defined, cannot deal with our data. It does not take into consideration the semantic aspect of the elements while comparing. For example if we compare the two dimension tables "film" and "movie" using the simple matching as defined in k-Mode, they will not be considered as the same, although they are two synonymous terms. So, we need to improve the dissimilarity measure by integrating other techniques such as the ontology, synonyms, etc.

More details about the assistant system are given in (Arfaoui and Akaichi, 2013).

3 THE USER REQUIREMENT

The requirement is a source of pertinent information. It plays a crucial role in the DW process design. It can cause the failure of the whole project if it is

Table 1: Comparison of different algorithms used to cluster categorical data.

Algorithm	Algorithm type	Complexity	Coefficient
K-MODE	Partitioning	O (n)	Simple Matching
ROCK	Hierarchical clustering	$O(kn^2)$	Links
QROCK	Hierarchical clustering	$O(n^2)$	Threshold
COOLCAT	Hierarchical clustering	$O\ (n^2)$	Entropy
LIMBO	Hierarchical clustering	O (nLogn)	Information Bottleneck
MULIC	Partitioning	$O\ (n^2)$	Hamming measure
HIERDENC	Hierarchical clustering	O (n)	Simple Matching
RAHCA	Hierarchical clustering	$O\ (An^3)$	New similarity measure based on Euclidean distance

faulty. It is used to specify “what data should be available and how it should be organized as well as what queries are of interest” (Malinowski and Zimanyi, 2008). It extracts the important elements related to the multidimensional schema (facts, measures, dimensions, attributes).

3.1 Collecting the Users’ Requirements

This step is about facilitating to the users the specification of their requirements. They can find difficulties to express their needs with the SQL queries especially when using the GROUP BY and/or HAVING clauses (Annoni et al., 2006), (Gyssens and Lakshmanan, 1997).

We propose, as solution, the use of an assistant system that intervenes by suggesting the possible multidimensional elements to use. The proposed system saves the manipulation of each user as a trace. Then, it exploits the stored experiences by making a set of comparisons between them and the current manipulation of the user. Finally, it suggests the appropriate elements to manipulate.

3.2 Generating the Schemas

To facilitate the manipulation of the users’ needs, we propose their presentation as star schemas (Figure 1). They have the following structure:

- Fact table corresponds to the subject of analysis. It is defined by **FN** and **MF{}** with:
  - **FN**: is the fact name “Profit”.
  - **MF {m1, m2, m3, m4, …}**: is the set of measures related to the fact F: “Quantity and Price”.
- Dimension tables represent the axis of analysis. Each one is composed by DN and A{}, with:
  - DN: is the dimension name: “Customer, Supplier, Product etc.”
  - A {a1, a2, a3, a4, …}: is the set of attributes describing the current dimension D: “FirstName, LastName, Address etc.”.

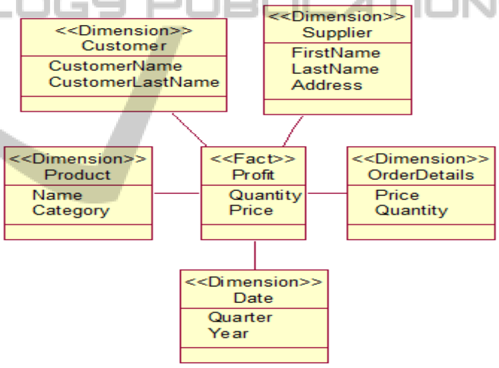


Figure 1: Example of schema corresponding to a user requirement.

4 THE CHARACTERISTICS OF K-MODE

The proposed algorithm is an extension of k-mode. This latter presents some challenges that we present in the following.

4.1 ‘k’: The Number of Clusters

One of the challenges related to the construct of clusters is the estimation of the appropriate ‘k’ (Hand et al., 2001). One of the proposed solutions is the calculation of the validity measure that is based on the inter-cluster and inter-cluster distance measures: validity= intra / inter (Malinowski and

Zimanyi, 2008). A second solution consists on using the gap statistic. It is about comparing the change in within-cluster dispersion with that expected under an appropriate reference null distribution (Tibshirani et al., 2001).

4.2 The Distance

Concerning the distance (Huang, 2008), (Ng et al., 2007), it is based on the simple matching dissimilarity measure that takes two values 0 or 1. It calculates the relative attribute frequencies of the cluster modes in the dissimilarity measure in the k-modes objective function. This modification allows the algorithm to recognize a cluster with weak intra-similarity, and therefore assigns less similar objects to such cluster, so that the generated clusters have strong intra-similarities.

4.3 The Selection of Modes

K-mode is unstable due to non-uniqueness of the modes (San et al., 2004). The clustering results depend strongly on the selection of modes during the clustering process. As solution, the authors propose “cluster center” represented by the most frequent values. Their formation is done using Cartisian product and union operations, and for the dissimilarity measure, it depends on the relative frequencies of categorical values within the cluster and simple matching between categorical values which can be considered as a categorical counterpart of the squared Euclidean distance measure.

5 THE PROPOSED ALGORITHM

5.1 The Specification of the New Algorithm

As we saw, the k-mode presents some disadvantages. To use it, we propose some improvements related to the number of clusters and the used distance.

- “k” the number of clusters: Since we are looking for grouping the different schemas into clusters, we suggest considering “k” as the number of existing domains. By this way, we are sure that the schemas of one cluster belong to same domain and they have some common elements. This choice facilitates the generation of the schemas of data mart.
- The distance: The k-mode uses the simple matching dissimilarity measure that is based on the relative frequencies of items within clusters. This measure cannot be applied to the schemas. As consequence, we propose its extension. The new distance will be detailed next.

5.2 The Ontology

In order to improve the simple matching dissimilarity measure, we propose the use of the ontology. In fact, the traditional measures, those are used for numerical data, categorical data and even for heterogeneous data, ignore the semantic knowledge. This has negatively influences on the quality of the interpretations (Batet et al., 2008), especially with the possibility to add semantic information about the domain in some fields (Studer et al., 1998).

Our ontology helps to extract the similar elements belonging to the same category. We need, then, the following classes:

- Domain: It corresponds to the domain of a specific schema.
- Schema: It identifies the schemas. It links its different elements.
- Fact: It corresponds to the subject of analysis. It includes all the different ways used to describe one fact.
- Measure: every fact has one or more measures that are numerical. We keep information about the different words used to describe a specific measure.
- Dimension: it corresponds to the axe of analysis. It serves to group the different ways to describe one specific dimension.
- Attribute: every dimension has a set of attributes. We keep information about the different words used to describe a specific attribute.

Once we specify the classes, we move to the relationships, and we have:

- is-Schema (Si, Dj): “Si” is a schema that belongs to the domain “Dj”.
- is-Fact (Fi, Sj): “Fi” is a fact that belongs to the schema “Sj”.
- is-Dimension (Di, Fj): “Di” is a dimension that belongs to the fact “Fj”.
- is-Measure (Mi, Fj): “Mi” is a measure that belongs to the fact “Fj”
- is-Attribute (Ai, Di): “Ai” is an attribute that is related to the dimension “Di”.

To exploit the ontology, we propose the implementation of a method that takes two elements to return three values: (-1), (0) and (1).

- (-1) implies that at least one of the two elements does not exist in the ontology.
- (0) implies that the two elements are not the same.
- (1) implies that the two elements are the same.

In the first case, we calculate the similarity of the elements of the two schemas taking into consideration the following points:

- The identical: It is the case where we use the same elements name in the two schemas. DeId (e1, e2) =1 if "e1" and "e2" are identical and 0 if not.
- The synonymous: It is the case where we use two different names that have the same meaning.

DeSy (e1, e2) = 1 if "e1" and "e2" are synonymous, and 0 if not.

- The typos: It is the case where the user makes mistakes when writing the name of the element. We calculate the degree of error. If it is low, we are in the case of typing error. If it is high we are in the case of two different words. In the following we only take into consideration the first case.

DeTy (e1, e2) =1 if "e1" and "e2" are the same with the existence of typing error.

- The post-fixe: It is the case where we use post-fixes to design the same thing.

DePost (e1, e2) = 1 if one of the two elements is the post-fixe of the other, and 0 if not.

- The pre-fixe: It is the case where we use pre-fixes to design the same thing.

DePre (e1, e2) = 1 if one of the elements is the pre-fixe of the other, and 0 if not.

Concerning the homonyms, it is taken into account implicitly. In fact, during the specification of the user requirements, we specify the domain. By this way, we adjust the terms so that two identical terms used in the same domain, denote the same thing and give the same information.

The degree of similarity of "e1" and "e2" (DeSim (e1, e2)) is measured by the numeric value in {0}, {1} using the formula (1).

$$\text{DeSim (e1, e2)} = [\text{DeId (e1, e2)} + \text{DeSy (e1, e2)} + \text{DeTy (e1, e2)} + \text{DePost (e1, e2)} + \text{DePre (e1, e2)}] \quad (1)$$

The result of this formula is '0' or '1'. '0' implies that the two elements are not similar, '1' if they are. The result is inserted into the ontology.

5.3 The New Dissimilarity Measure

Since we are dealing with star schema, we have as elements: fact table, dimension tables, measures and attributes.

The new measure Coef_{SM} representing the coefficient of similarity is presented through the following formula (2):

$$\text{Coef}_{\text{SM}} = [\,(\text{MaxD} - \text{CoefD}) / \text{MaxD}] + [(\text{MaxM} - \text{CoefM}) / \text{MaxM}] + [(\text{MaxF} - \text{CoefF}\,) / \text{MaxF}] + [\,(\,\text{MaxA} - \text{CoefA}\,) / \text{MaxA}] \quad (2)$$

With:

- MaxD: corresponds to the maximum number of the existing dimension tables.
- MaxM: corresponds to the maximum number of the existing measures.
- MaxF: corresponds to the maximum number of the existing fact tables.
- MaxA: corresponds to the maximum number of the existing attributes.
- CoefD: calculates the number of similar dimension tables using the formula (1).
- CoefM: calculates the number of similar measures using the formula (1).
- CoefF: calculates the number of similar fact tables using the formula (1).
- CoefA: calculates the number of similar attributes using the formula (1).

5.4 The Algorithm

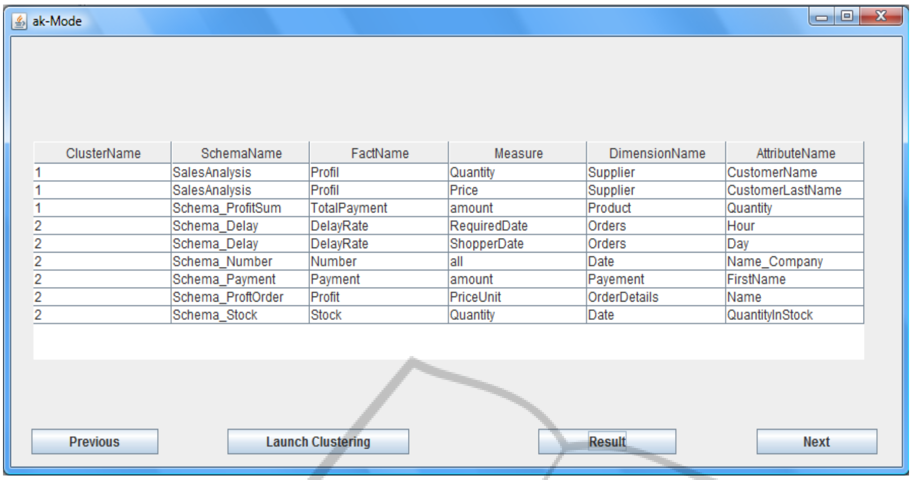
The new algorithm has the same steps as k-mode:

a) Define the 'k' number of existing domains
b) Select 'k' initial modes.
c) Allocate a schema to the cluster whose mode is the nearest to the cluster, using the formula (2):
d) Update the mode of the cluster after each allocation.
e) After all schemas have been allocated to the respective cluster, retest the schemas with new modes and update the clusters.
f) Repeat steps (b) and (c) until there is no change in clusters.

Concerning the update of the mode, we use the same method as defined in the k-mode which is based on the frequency of the elements.

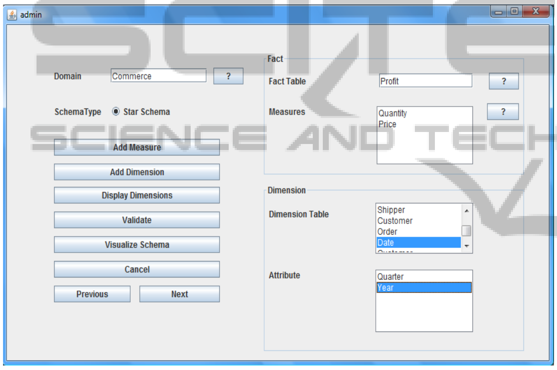
6 THE IMPLEMENTATION

In this section, we present the implementation of our solution. Figure 2 presents the interface that we use to collect the users' requirements.



ClusterName	SchemaName	FactName	Measure	DimensionName	AttributeName
1	SalesAnalysis	Profil	Quantity	Supplier	CustomerName
1	SalesAnalysis	Profil	Price	Supplier	CustomerLastName
1	Schema_ProfitSum	TotalPayment	amount	Product	Quantity
2	Schema_Delay	DelayRate	RequiredDate	Orders	Hour
2	Schema_Delay	DelayRate	ShopperDate	Orders	Day
2	Schema_Number	Number	all	Date	Name_Company
2	Schema_Payment	Payment	amount	Payement	FirstName
2	Schema_ProftOrder	Profit	PriceUnit	OrderDetails	Name
2	Schema_Stock	Stock	Quantity	Date	QuantityInStock

Figure 5: The result of the application of the new algorithm.



Domain: Commerce

SchemaType: ● Star Schema

Fact

Fact Table: Profit

Measures: Quantity, Price

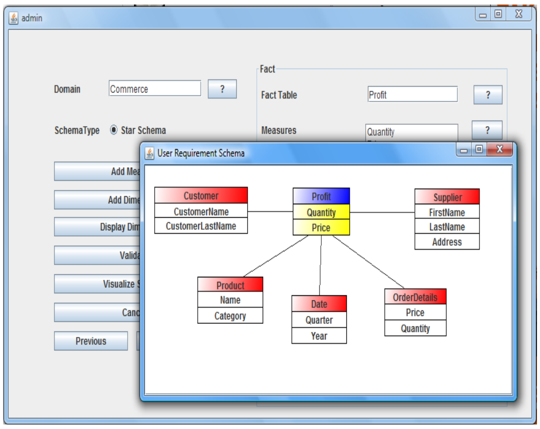
Dimension

Dimension Table: Shipper, Customer, Order, Date

Attribute: Quarter, Year

Figure 2: The proposed interface to specify the users' requirements.

Once he finishes all the specifications, the result of this step is visualized as a star schema as presented in Figure 3. Such modeling facilitates the task of clustering.



Domain: Commerce

SchemaType: ● Star Schema

Fact

Fact Table: Profit

Measures: Quantity

Figure 3: Example of star schema corresponding to the user requirement.

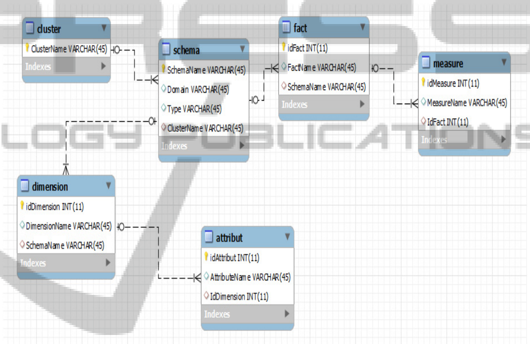


Figure 4: The structure of the used database.

The collected schemas are stored into database (Figure 4). It is composed by 6 classes which are "Cluster", "Schema", "Fact", "Measure", "Dimension" and "Attribute".

Once we collect the set of schemas corresponding to the users' requirements, we move to the next step where we apply our new algorithm to cluster them. The result of the clustering is presented in Figure 5 where we have two clusters. For each cluster, it visualizes the set of existing elements.

7 CONCLUSION

In this work, we proposed a new algorithm ak-mode to cluster the schemas that were generated from the users' requirements. The new algorithm is an extension of k-mode algorithm. It is chosen because of its capacity to deal with huge amount of data; also, it has the lowest temporal complexity. The new algorithm uses the ontology to improve the quality of the dissimilarity measure to take into consideration the semantic aspect while comparing the elements of the schemas. The result of this algorithm is a set of clusters containing a set of schemas semantically close.

The new algorithm offers the possibility to deal with the requirements of different users having different skills and belonging to different departments.

As future work, we will merge the schemas within each cluster using the schema integration technique to generate data mart schemas.

We propose, also, extending this work to deal with other structures of schemas corresponding to the databases schemas.

ACKNOWLEDGEMENTS

We would like to thank everyone.

REFERENCES

Alexander, J. H., Freiling, M. J., Shulman, S. J., Staley, J. L., Rehfuss, S., and Messick, S. L., 1986. Knowledge Level Engineering: Ontological Analysis. *In Proceedings of the 5th National Conference on Artificial Intelligence, AAAI-86, 963-968.*

Alexiev, V., Breu, M., De Bruijn, J., Fensel, D., Lara, R., and Lausen, H., 2005. *Information Integration with Ontologies: Experiences from an Industrial Showcase,* John Wiley & Son.

Andreopoulos, B., An, A., and Wang, X., 2004. *MULIC: Multi-Layer Increasing Coherence Clustering of Categorical data sets.* Technical Report CS-2004-07, York University.

Andritsos, P., Tsaparas, P., Miller, R. J., and Sevcik, K. C., 2004. LIMBO: Scalable Clustering of Categorical Data. *In Proceedings of the 9th International Conference on Extending Database Technology (EDBT), Heraklion, Greece, 123-146.*

Annoni, E., Ravat, F., Teste, O., and Zurfluh, G., 2006. Towards Multidimensional Requirement Design. *In Proceedings of 8th International Conference Data Warehousing and Knowledge Discovery (DaWaK), 75-84.*

Arfaoui. N., Akaichi. J., 2013. New Approach for the Collection of Users' Requirements using DwADS. In Proceedings of 22nd International Business Information Management Association (IBIMA), Rome, Italy,

Barbara, D., Couto, J., and Li, Y., 2002. COOLCAT: An entropy-based algorithm for categorical clustering. In Proceedings of the eleventh international conference on Information and knowledge management, 582-589.

Batet, M., Valls, A., and Gibert, K.., 2008. Improving classical clustering with ontologies. In Proceedings of the 4th world conference of the international association for statistical computing, 137-146.

Chavent, M., Kuentz, V., and Saracco, J., 2010. *Clustering of categorical variables around latent variables*. Cahiers du GREThA 2010-02, Groupe de Recherche en Economie Theorique et Appliquee.

Chen, D., Cui, D.W., Wang, C.X., and Wang, Z. R., 2006. A Rough Set-Based Hierarchical Clustering Algorithm for Categorical Data. *International Journal of Information Technology.*

Faber, V., 1994. Clustering and the Continuous k-means Algorithm. *Los Alamos Science, 138-144*.

Guha, S., Rastogi, R., and Shim, K.., 2000. ROCK: A Robust Clustering Algorithm for Categorical Attributes. *In: Inf. Syst., Vol. 25, Nr. 5 Oxford, UK, UK: Elsevier Science Ltd., 345-366.*

Gyssens, M. and Lakshmanan, L. V. S., 1997. A Foundation for Multi-dimensional Databases. In Proceedings of 23rd International Conference on Very Large Data Bases (VLDB), 106-11.

Hand, D., Mannila, H., and Smyth, P., 2001. Principles of Data Mining. *MIT Press, Cambridge, MA*.

Hotho, A., Staab, S., and Stumme, G., 2003. Wordnet improves Text Document Clustering. *In Proceedings of the SIGIR 2003 Semantic Web Workshop*.

Huang, Z., 1998. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Data Mining and Knowledge Discovery, 2:283–304.*

Jain, A. K., Murty, M. N., and Flynn, P. J., 1999. Data Clustering: A Review. *ACM Comput. Surv., 264-323.*

Jing, L., Zhou, L., Ng, M. K., and Huang, J. Z., 2006. Ontology-based Distance Measure for Text Clustering. *In Proceeding of SIAM International conference on Text Data Mining, Bethesda*.

Khan, S. S., Kant, S., 2007. Computation of Initial Modes for K-modes Clustering Algorithm using Evidence Accumulation. *International Joint Conference on Artificial Intelligence, 2785-2789.*

Malinowski, E., and Zimanyi, E., 2008. Advanced Data Warehouse Design, From Conventional to Spatial and Temporal Applications, Springer Verlag Berlin Heidelberg.

Ng, M. K., Li, M. J., Huang, J. Z., and He, Z., 2007. On the Impact of Dissimilarity Measure in k-modes Clustering Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (3): 503-507.*

Quine, W.V.O., 1980. *From a Logical Point of View.* Harvard University Press; Cambridge, MA.

Rezankova, H., 2009. Cluster Analysis and Categorical Data. *Statistika, 216-232*.

San, O. M., Huynh, V. N., and Nakamori, Y., 2004. An Alternative Extension Of The K-Means Algorithm For Clustering Categorical Data. *Journal of Applied Mathematics and Computer Science, No. 2, 241-247.*

Studer, R., Benjamins, V. R., and Fensel, D., 1998. Knowledge Engineering: Principles and Methods. *IEEE Trans on Data and Knowledge Engineering, 25 (1-2): 161-197.*

Tibshirani, R., Walther, G.., and Hastie, T., 2001. Estimating the number of clusters in a data set via the gap statistic, *J. R. Statist. Soc. B, 411-423.*