A Novel Clustering Algorithm to Capture Utility Information in Transactional Data

Piyush Lakhawat, Mayank Mishra and Arun Somani
Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa, U.S.A.

Keywords: Clustering Algorithm, Transactional Data, High Utility Patterns.

Abstract: We develop and design a novel clustering algorithm to capture utility information in transactional data. Transactional data is a special type of categorical data where transactions can be of varying length. A key objective for all categorical data analysis is pattern recognition. Therefore, transactional clustering algorithms focus on capturing the information on high frequency patterns from the data in the clusters. In recent times, utility information for category types in the data has been added to the transactional data model for a more realistic representation of data. As a result, the key information of interest has become high utility patterns instead of high frequency patterns. To the best our knowledge, no existing clustering algorithm for transactional data captures the utility information in the clusters found. Along with our new clustering rationale we also develop corresponding metrics for evaluating quality of clusters found. Experiments on real datasets show that the clusters found by our algorithm successfully capture the high utility patterns in the data. Comparative experiments with other clustering algorithms further illustrate the effectiveness of our algorithm.

1 INTRODUCTION AND MOTIVATION

Transactional data model is used in many important applications of data mining and analytics like market basket analysis, bioinformatics, click stream analysis etc. Clustering is one of key analysis techniques for these applications. For example, clustering of customer transactions data is performed in market basket analysis for segmentation and identification of various customer types (Ngai et al., 2009). An important data type in bioinformatics experiments is gene co-expression data which can be modeled as transactional data (Andreopoulos et al., 2009). Identifying clusters of genes exhibiting similar expression in controlled conditions can lead to critical discoveries in medicine.

A typical transaction of size k in a transactional data set can represented as $T_{ID} = (X_1, X_2 \ldots X_K)$, where X_i is a category type(also called item type) and ID is the transaction ID. For example, transactions from a retail store data can be like T_1 =(Candy, Pop, Chips), T_2 =(HDTV, Hi-Definition Speakers) and so on, where each transaction is a list of items bought by a customer in one trip to the store. If we work with only this level of information then we assume equal importance of each category type in the analysis. This would mean that if more number of customer purchase patterns include Pop than HDTVs then our analysis would implicitly assume Pop being a more important category type. However, due to significantly more profit per sale of a HDTV, it is possible that overall HDTV sales yield much more profit to the store than Pop sales. The general idea is that various category types in a transactional data have different level of relative importance (called utility) in the analysis. Therefore to make the transactional data model more realistic utility values are assigned to each category type in the data. An important data mining problem which also uses transactional data model is itemset mining. The aforementioned idea has led to the evolution of traditional frequent itemset mining problem into the current high utility itemset mining problem (Tseng et al., 2015). Such a transition has not yet occurred for the transactional data clustering problem. To the best of our knowledge no existing clustering algorithm for transactional data captures the utility information.

Current transactional clustering algorithms capture the frequency information (number of occurrences) while finding clusters (Yan et al., 2010) but do not use the utility values. This will lead to misleading clusters especially when various category types have significant difference in their utility. In order to cap-

Lakhawat, P., Mishra, M. and Somani, A.
A Novel Clustering Algorithm to Capture Utility Information in Transactional Data.
DOI: 10.5220/0006092104560462
In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR*, pages 456-462
ISBN: 978-989-758-203-5

ture high utility patterns (patterns based on combination of frequency and utility) in the data, we propose a concept of relative utility of category types in clusters. We use it to define a new similarity metric to guide the clustering process. Based on our similarity metric we develop a hierarchical agglomerative clustering algorithm. The algorithm design allows two tunable parameters to be set to achieve clusters with desired properties. Since we are introducing a new clustering rationale, we propose two corresponding cluster evaluation criterion as the existing ones dont take utility values into account.

1.1 Contributions

Our work makes the following contributions:

- A novel clustering algorithm to capture utility information in transactional data. The algorithm can be tuned to obtain clusters with desired properties of a strong core and a balanced structure.
- Two validation criterion for evaluating the obtained cluster structure based on how accurately it captures the high utility patterns in the data.
- Experimental results on real data sets showing that the clustering algorithm successfully captures the high utility patterns in the data. Comparative experiments with other algorithms illustrate the effectiveness of our algorithm.

2 PRIOR WORK ON CATEGORICAL AND TRANSACTIONAL CLUSTERING

Transactional data is a special type of categorical data where transactions can be of varying lengths. While it is possible to use categorical clustering algorithms for transactional data, most of them lead to a data explosion problem as described in (Yan et al., 2010). Specific algorithms targeting transactional data clustering have been developed as well (Wang et al., 1999), (Yan et al., 2005), (Yan et al., 2010), (Yang et al., 2002). Use of similarity measures and cluster quality measures are two common approaches used in most of the categorical and transactional clustering algorithms. Based on our review of the published work we classify them into the following three categories:

- **Similarity Measures.** In (Huang, 1998) an extension of the popular k-means algorithm (which is for numerical data) is presented called k modes. It replaces the means with modes of clusters to minimize a cost function. An in-depth analysis of subsequent updates of the k-modes algorithm is done in (Bai et al., 2013). In (Guha et al., 1999) a link based similarity measure is proposed to guide the clustering. An entropy based similarity criterion is presented in (Chen and Liu, 2005). A transformation technique is presented in (Qian et al., 2015) to map categorical data into a space structure followed by using similarity measures. A similarity measure based using information bottleneck theory is presented in (Tishby and Slonim, 2000) and (Andritsos et al., 2004). (Ganti et al., 1999), (Gibson et al., 1998) and (Wang et al., 1999) are some other works with the focus on similarity type measures for clustering. A mapping of categorical data into a tree structure is performed in (Chen et al., 2016) before performing clustering using a tree similarity metric.
- **Quality Measures.** (Yang et al., 2002), (Yan et al., 2005), (Barbará et al., 2002) and (Yan et al., 2010) present algorithms which iteratively improve overall cluster quality. These definitions are based on various entropy based concepts from information theory. An in-depth discussion of entropy based criterion used for the categorical clustering is presented in (Li et al., 2004).
- **Miscellaneous.** Categorical clustering has been presented as an optimization problem in (Xiong et al., 2012). A learning based multi-objective fuzzy approach is presented in (Saha and Maulik, 2014). An entropy based subspace clustering technique for categorical data is presented in (Sun et al., 2015).

While the current body of work has numerous clustering algorithms, all lack of capturing the utility information in the data. We propose the following algorithm to fill the void.

3 A NOVEL CLUSTERING ALGORITHM TO CAPTURE UTILITY INFORMATION IN TRANSACTIONAL DATA

Any clustering algorithm is designed with a clustering criterion at its core. The clustering criterion behind our algorithm is to gather transactions which share common high utility category types in a cluster. At the convergence of clustering we expect to capture high utility patterns of the data. While the current algorithms for transactional clustering aim to capture high frequency patterns while clustering, we argue that this

can lead to inaccurate clustering in cases where the utilities of category types have a steep distribution. A high utility cluster (containing transactions with common high utility category types) can be missed (i.e. not found) if we focus only on frequency information while clustering. Before further describing our clustering algorithm in detail we need to define the following preliminaries (common in utility itemset mining (Tseng et al., 2015)):

$$I = \{a_1, a_2, \ldots, a_M\} = \text{Set of item(category) types} \quad (1)$$

$$D = \{T_1, T_2, \ldots, T_N\} = \text{Transactional dataset} \quad (2)$$

where each $T_i = \{x_1, x_2, \ldots\} \subset I$

$$X = \text{size k itemset} = \{x_1, x_2, \ldots, x_k | x_i \in I\} \quad (3)$$

$$eu(a_i) = \text{ external utility of } a_i \quad (4)$$

$$iu(a_i, T_j) = \text{ internal utility of } a_i \text{ in } T_j \quad (5)$$

$eu(a_i)$ is the measure of unit importance for item type a_i. While $iu(a_i, T_j)$ is a quantity measure of a_i in T_j. The absolute utility of an item in a transaction is defined as the product of its internal and external utility.

$$au(a_i, T_j) = eu(a_i).iu(a_i, T_j) \quad (6)$$

Absolute utility of an itemset in a transaction is the sum of absolute utilities of its constituent items.

$$au(X, T_j) = \sum_{x_i \in X} au(x_i, T_j) \quad (7)$$

Absolute utility of a transaction (also called transaction utility) is the sum of absolute utilities of all its constituent items.

$$TU(T_j) = \sum_{x_i \in T_j} au(x_i, T_j) \quad (8)$$

Absolute utility of an itemset in the dataset D is the sum of absolute of that itemset in all transactions that it occurs in.

$$au(X, D) = \sum_{X \in T_j \wedge T_j \in D} au(X, T_j) \quad (9)$$

The set of HUI in D is the collection of all itemsets which have absolute utility more than or equal to ϕ in the dataset D.

$$\text{HUI in D} = \{X \text{ such that } au(X, D) \geq \phi\} \quad (10)$$

A cluster C_k of is a subset of transactions from D.

$$C_k = \{T_1, T_2 \ldots T_k | T_i \in D\} \quad (11)$$

C is the set of all given clusters.

$$I_{C_k} = \{a_i | a_i \in T_j \wedge T_j \in C_k\} = \text{item types in } C_k \quad (12)$$

We introduce following **new concepts** of cluster utility, relative utility of a category type in a cluster and the affinity between clusters.

$$CU(C_k) = \sum_{T_j \in C_k} TU(T_j) = \text{Cluster utility of } C_k \quad (13)$$

We define cluster utility as an overall measure of importance of a cluster, since it is the sum of utilities of all transactions in it.

$$\forall a_i \in I_{C_k}, ru(a_i, C_k) = \frac{\sum_{a_i \in I_{C_k} \wedge T_j \in C_k} au(a_i, T_j)}{CU(C_k)} \quad (14)$$

$$= \text{relative utility of } a_i \text{ in } C_k$$

We define relative utility as the relative importance (since utility is a unit of importance) given to a_i among all I_{C_k} in C_k. The set $\{ru(a_i, C_k) | a_i \in I_{C_k}\}$ is then a of signature of the cluster. It is a concise representation of the utility information we are interested in about the cluster. And the collection of all such sets gives us the concise global information of the cluster structure present in the dataset on the basis of high utility patterns in it.

For clusters C_i and C_j:

$$affinity(C_i, C_j) = \sum_{a \in I_{C_k} \wedge a \in I_{C_j}} min(ru(a, C_i), ru(a, C_j)) \quad (15)$$

We define affinity as the sum of shared utility of common category types among two clusters. So it is a representative of the similar importance given by two clusters to their common category types. This aligns with our goal of clustering which is to gather transactions which share common high utility category types in a cluster.

3.1 Clustering Algorithm

Using our definition of affinity we perform clustering on the dataset using a bottom-up (also called agglomerative) hierarchical approach. Agglomerative hierarchical approach is well established for successful clustering of categorical data(Guha et al., 1999). We initialize our algorithm by assigning each transaction as an individual cluster. We then calculate affinity of each cluster with every other cluster. Following that we incrementally merge the two clusters which have most affinity to each other. A merge implies union of the two clusters to form a single cluster. After each merge we calculate all the affinity values associated with the new cluster. This includes first computing the cluster utility and relative utilities for the new cluster.

The cluster structure represents a complete graph with each cluster as a node connected to every other node. The node weights represent the cluster utility and the edge weights represent the affinity values. At

each merge we calculate the node weight for the new node and the edge weights for the new edges while keeping the graph complete. We terminate the algorithm when the maximum affinity found between any two clusters is below a certain predefined affinity threshold min_{aff}. The cluster distribution can have a long tail, meaning there can be many transactions which do not fall under any significant cluster structure. min_{aff} ensures that clusters with high cluster utility do not end up merging just because all other remaining small clusters are very dissimilar among each other.

At this point we only select clusters with more than or equal to a minimum cluster utility value. We do this by predefining a parameter min_{uty} and accepting only those clusters which have cluster utility more than or equal to $min_{uty}{}^{th}$ of the cluster with the maximum cluster utility. Figure 1 shows a small synthetic example illustrating the clustering process. It shows affinities and cluster utilities as edge and node weights respectively. The change in cluster structure can be observed based on the choice of min_{aff} and the min_{uty} parameters. Algorithm 1 below provides a pseudocode.

Input: C ;
while $max_{aff} \geq min_{aff}$ **do**
  for $C_i, C_j \in C$ **do**
    if $affinity(C_i, C_j) > max_{aff}$ **then**
      max_{aff} = affinity(C_i, C_j);
      $C_{m1} = C_i$;
      $C_{m2} = C_j$;
  merge(C_{m1}, C_{m2});
  update relevant affinities;
for $C_t \in C$ **do**
  if $\frac{CU(C_t)}{max(CU(C_k) \forall C_k \in C)} \leq min_{uty}$ **then**
    delete C_t;
return C;

Algorithm 1: A novel clustering algorithm for categorical data with utility information.

4 PROPERTIES OF THE CLUSTER STRUCTURE

The obtained cluster structure by our algorithm will have the following properties by design:

- A strong core: We terminate the clustering process when the maximum affinity found between any two clusters is less than min_{aff}. Therefore, each cluster will have a set of category types for which each transaction in the cluster has at least min_{aff} of shared relative utility among them. This is evident from the definition of affinity.
- A balanced structure: Ratio of cluster utility of each cluster to that of the cluster with the maximum cluster utility will be more than or equal to the predefined threshold min_{uty}.
- Greedy path taken: Clusters with the most affinity with each other merge at each step.

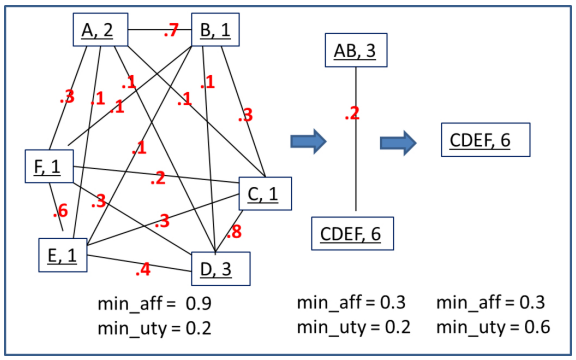

Figure 1: Illustration of the clustering process.

Besides the above three properties, we expect the cluster structure to capture the high utility patterns in the data. This is the primary goal of this work. A recent transactional clustering framework SCALE(Yan et al., 2010) introduces a cluster validation criterion called LISR(Large Item Size Ratio) which captures large items(frequently occurring category types) preserved in clustering. Since we capture patterns based on combination criterion of frequency and utility, using validation criterion like LISR is inadequate. Therefore we introduce two new validation criterion based on the high utility itemsets (HUI) present in the clusters. We call them $HUI_{capture}$ and $HUI_{inaccurate}$. They are defined as follows:

$$\text{HUI in C} = \{X \text{ such that } au(X, C_k) >= \phi' | C_k \in C\} \quad (16)$$

$$\text{where } \phi' = \frac{\phi}{\frac{|D|}{\Sigma_{C_k \in C} |C_k|}}.$$

The set of HUI in C represents the high utility patterns present in the cluster structure C. If the entire data set D is used in clustering then $\phi = \phi'$. However, since clustering is an expensive operation often random sampling techniques are used to select transactions for clustering. In those cases we scale down the threshold ϕ to ϕ'.

$$HUI_{capture} = \frac{|\{\text{HUI in C}\} \cap \{\text{HUI in D}\}|}{|\{\text{HUI in D}\}|} \quad (17)$$

$HUI_{capture}$ represents the degree to which the cluster structure C captures the high utility patterns

present in the dataset D. Successful $HUI_{capture}$ is characterized by a value greater than the ratio of size of data used for clustering to the size of the whole dataset $\frac{\sum_{C_k \in C} |C_k|}{|D|}$. Higher values of $HUI_{capture}$ imply higher quality of the cluster structure.

$$HUI_{inaccurate} = \frac{|\{\text{HUI in C}\} - \{\text{HUI in C} \cap \text{HUI in D}\}|}{|\{\text{HUI in D}\}|} \quad (18)$$

$HUI_{inaccurate}$ represents the degree of inaccurately captured high utility patterns in C with respect to the high utility patterns present in the data D. Lower values of $HUI_{inaccurate}$ imply higher quality of the cluster structure.

5 EXPERIMENTAL EVALUATION AND DISCUSSION

To validate our algorithm we perform clustering on a real data set obtained from (Ret, 2003) and provided by (Brijs et al., 1999). It contains anonymous customer transaction data from a Belgian retail store and contains 88,163 transactions. We randomly generated the external utilities (between 1-50) for various category types by using a uniform random number generator. Using this data we performed the following experiment:

- For ϕ=50000 we calculated the list of high utility itemsets (HUI) in the complete data set D. We found 111 HUIs. We did this by implementing a popular itemset mining technique called the two-phase algorithm (Liu et al., 2005). The choice of ϕ was based on obtaining a reasonably large but manageable number of HUI.
- For four combinations of our algorithm parameters we calculated the cluster structure and the list of HUI for each cluster structures. The selection of data for clustering is done using uniform random sampling.
- Finally, we evaluate our validation metrics: $HUI_{capture}$ and $HUI_{inaccurate}$. We present the results in Figure 2.

The following inferences can be drawn from the results:

- The HUI captured in clusters are significantly greater than the data used for clustering. This implies high quality of the cluster structure.
- The extremely low (and 0) values of HUI inaccurately captured imply high accuracy of the cluster structure.

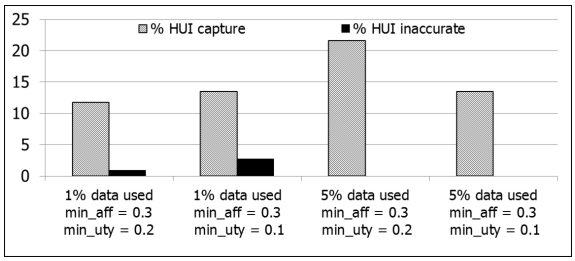

Figure 2: Cluster validations metrics for the retail store data set.

Table 1: Comparative results for Soybean dataset: patterns missed.

ϕ	Our Algorithm	BKPlot	K-modes
800	0	0	8
400	2	5	7
200	0	2	4

We also perform comparative experiments with two other categorical clustering algorithms: K-modes (Huang, 1998) and BKPlot (Chen and Liu, 2005). Since the provided implementations of K-modes (Kmo, 2015) and BKPlot (BKP, 2008) accept data sets only with uniform transaction length, we perform the comparative analysis on the Soybean data set obtained from (UCI, 1987). It is a collection of attributes of various soybean plants type. Soybean data set is reasonably small so we dont use the random sampling and instead perform clustering on the entire dataset. To maintain uniformity in cluster structure we evaluate a four cluster solution for each algorithm. For using in our algorithm we also generate utility values(between 1-50) for each category type using a uniform random number generator. Since the data set is small and we dont use random sampling, we perform comparison based on patterns (high utility itemsets) missed by clusters formed by each algorithm. Table 1 shows the comparative results. ϕ represents the threshold value used to calculate high utility patterns (itemsets). Our algorithm performs better (or equal) than the other two for all cases. While the soybean data set is a small one, it still illustrates the fact that well established algorithms like K-modes and BKPlot miss out on high utility patterns.

6 CONCLUSIONS AND FUTURE WORKS

In the growing body of work of clustering algorithms for categorical data we identified the need for an algorithm which captures the utility information in the transactional data. The current algorithms for trans-

actional clustering aim to capture high frequency patterns while clustering. We argue that this can result in misleading clusters especially for cases where the utilities of category types have a steep distribution. High utility clusters can be missed if we focus only on frequency information while clustering. We propose a novel clustering algorithm for transactional data which captures the utility information. We also propose two validation criterion for the obtained cluster structure based on how accurately it captures the high utility patterns in the data.

Our experiments on a real data sets show that the clustering algorithm successfully captures the high utility patterns in the data. Our comparative experiment results further illustrate the effectiveness of our algorithm over the popular K-modes and BKPlot algorithms. For future work, we plan to do experiments on data sets from various applications like bioinformatics, click stream data etc. We believe interpretations of clusters found in different data sets will lead to interesting results and evolution of our algorithm. We plan to develop the idea of utility aware clustering for entropy based clustering methods as well.

ACKNOWLEDGEMENTS

The research reported in this paper is funded in part by Philip and Virginia Sproul Professorship Endowment and Jerry R. Junkins Endowments at Iowa State University. The research computation is supported by the HPC@ISU equipment at Iowa State University, some of which has been purchased through funding provided by NSF under MRI grant number CNS 1229081 and CRI grant number 1205413. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

REFERENCES

(1987). Uci machine learning repository. archive.ics.uci.edu/ml/datasets/Soybean+(Small). Accessed: 2016-09-01.

(2003). Frequent itemset mining dataset repository. http://fimi.ua.ac.be/data/. Accessed: 2016-06-14.

(2008). Bkplot implementation. cecs.wright.edu/~keke.chen/. Accessed: 2016-09-01.

(2015). K-modes implementation. https://github.com/nicodv/kmodes. Accessed: 2016-09-01.

Andreopoulos, B., An, A., Wang, X., and Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, 10(3):297–314.

Andritsos, P., Tsaparas, P., Miller, R. J., and Sevcik, K. C. (2004). Limbo: Scalable clustering of categorical data. In *International Conference on Extending Database Technology*, pages 123–146. Springer.

Bai, L., Liang, J., Dang, C., and Cao, F. (2013). The impact of cluster representatives on the convergence of the k-modes type clustering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1509–1522.

Barbará, D., Li, Y., and Couto, J. (2002). Coolcat: an entropy-based algorithm for categorical clustering. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 582–589. ACM.

Brijs, T., Swinnen, G., Vanhoof, K., and Wets, G. (1999). Using association rules for product assortment decisions: A case study. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 254–260. ACM.

Chen, K. and Liu, L. (2005). The" best k" for entropy-based categorical data clustering.

Chen, X., Huang, J. Z., and Luo, J. (2016). Purtreeclust: A purchase tree clustering algorithm for large-scale customer transaction data. In *32nd IEEE International Conference on Data Engineering*, pages 661–672. IEEE.

Ganti, V., Gehrke, J., and Ramakrishnan, R. (1999). Cactusclustering categorical data using summaries. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 73–83. ACM.

Gibson, D., Kleinberg, J., and Raghavan, P. (1998). Clustering categorical data: An approach based on dynamical systems. *Databases*, 1:75.

Guha, S., Rastogi, R., and Shim, K. (1999). Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304.

Li, T., Ma, S., and Ogihara, M. (2004). Entropy-based criterion in categorical clustering. In *Proceedings of the twenty-first international conference on Machine learning*, page 68. ACM.

Liu, Y., Liao, W.-k., and Choudhary, A. (2005). A two-phase algorithm for fast discovery of high utility itemsets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 689–695. Springer.

Ngai, E. W., Xiu, L., and Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, 36(2):2592–2602.

Qian, Y., Li, F., Liang, J., Liu, B., and Dang, C. (2015). Space structure and clustering of categorical data.

Saha, I. and Maulik, U. (2014). Incremental learning based multiobjective fuzzy clustering for categorical data. *Information Sciences*, 267:35–57.

Sun, H., Chen, R., Jin, S., and Qin, Y. (2015). A hierarchical clustering for categorical data based on holo-entropy. In *2015 12th Web Information System and Application Conference (WISA)*, pages 269–274. IEEE.

Tishby, N. and Slonim, N. (2000). Data clustering by markovian relaxation and the information bottleneck method. In *NIPS*, pages 640–646. Citeseer.

Tseng, V. S., Wu, C.-W., Fournier-Viger, P., and Philip, S. Y. (2015). Efficient algorithms for mining the concise and lossless representation of high utility itemsets. *IEEE transactions on knowledge and data engineering*, 27(3):726–739.

Wang, K., Xu, C., and Liu, B. (1999). Clustering transactions using large items. In *Proceedings of the eighth international conference on Information and knowledge management*, pages 483–490. ACM.

Xiong, T., Wang, S., Mayers, A., and Monga, E. (2012). Dhcc: Divisive hierarchical clustering of categorical data. *Data Mining and Knowledge Discovery*, 24(1):103–135.

Yan, H., Chen, K., Liu, L., and Yi, Z. (2010). Scale: a scalable framework for efficiently clustering transactional data. *Data mining and knowledge Discovery*, 20(1):1–27.

Yan, H., Zhang, L., and Zhang, Y. (2005). Clustering categorical data using coverage density. In *International Conference on Advanced Data Mining and Applications*, pages 248–255. Springer.

Yang, Y., Guan, X., and You, J. (2002). 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*, pages 682–687. ACM.