A Simple Yet Fast Clustering Approach for Categorical Data

Garima Khandelwal
M.Tech Scholar,CSE
RCEW,Jaipur

Rakesh Sharma
Assistant Professor, CSE
RCEW, Jaipur

ABSTRACT
Categorical data has always posed a challenge in data analysis through clustering. With the increasing awareness about Big data analysis, the need for better clustering methods for categorical data and mixed data has arisen. The prevailing clustering algorithms are not suitable for clustering categorical data majorly because the distance functions used for continuous data are not applicable for categorical data. Recent research focuses on several different approaches for clustering categorical data. However, the complexity of methods makes them unsuitable for use in big data. Emphasis should be on algorithms which are faster. Thus paper proposes a simple, fast method derived from statistics for clustering categorical data. Results on popular datasets are encouraging.

General Terms
Data Mining, Supervised learning, Clustering.

Keywords
Clustering, categorical data, big data, k-means.

1. INTRODUCTION
Clustering is a widely used tool which has applications ranging from data mining, Big data analytics to machine learning. Clustering methods have been researched thoroughly since decades and yet are developing due to changing requirements and open challenges. Conventional and established clustering methods like k-means [1, 2, 3] have been found to work well only for numeric data. Non-numeric data which is further classified as categorical, ordinal etc. is still difficult to handle. Since computations are easier to perform and translate into programs when all data is numeric, the main technique to handle categorical data is to first convert it into equivalent numeric value and then apply a conventional clustering algorithm. Many approaches have been proposed for this conversion, which can be broadly categorized as direct, dissimilarity-based, fuzzy set approach, context based etc.

Natural process of clustering aims at collecting similar items together, hence, measures of similarity and dissimilarity have always been a research area. Finding similarity between numeric data is easier and handles through ideas derived from statistics and geometry. But this is difficult to adapt for categorical data due to many reasons: no natural order, high dimensionality and existence of subspace clusters.

This paper proposes a simple method to convert categorical data into numeric so that only a small pre-processing step is required, and can be followed by a conventional clustering method. Experiments over categorical datasets have been performed using a variant of k-means; results exhibit effectiveness of the proposed method.

2. CLUSTERING CATEGORICAL DATA - SURVEY
Categorical datasets impose a number of challenges on clustering methods, the most significant of which is the lack of a natural order on the individual domains. Due to this a large number of traditional similarity measures become ineffective. Generally, these measures are based on co-occurrence of attribute values. The popular measures like Jaccard coefficient, cosine similarity may lead similarity to be defined even between attribute values that never occur together for any data point. Secondly, categorical datasets are generally high dimensional, though, this is not a direct consequence. Hence, it is suggested that clustering approaches for categorical data should be highly scalable in terms of number of attributes. In high-dimensions, it can be shown that traditional distance measures become ineffective, a phenomenon known as the curse of dimensionality [4]. Finally, many categorical datasets suffer from subspace clusters, that is, they do not exhibit clusters over all dimensions rather on a subset of dimensions. A classic example is document clustering, where though the entire dictionary is very large but individual documents contain relatively few words. Thus it may be desirable to identify clusters in subspaces.

The above discussion entails a number of key characteristics for good categorical clustering algorithms, which should ideally not impose any constraints or assumptions on the underlying domain, scale well over the number of attributes, and detect clusters not only over all attributes, but also over subsets thereof.

A major step towards categorical clustering was taken by Huang [5] through modification of well-known k-means into k-modes algorithm. The k-means algorithm works very well for numeric data. The notion of cluster representatives by mean values was replaced by a notion of modes for the categorical data. The algorithm preserves scalability of the k-means algorithm but also inherits its drawback of dependence on initialization.

STIRR presented by Gibson et al. [6] maps datasets into a hypergraph structure of weighted vertices that correspond to individual attribute values. STIRR iterates multiple instances (so-called basins) of these graphs using a user defined combination operator to eventually converge to a fix point. Upon convergence, the weights of the basins can be used to partition the data points, yielding the final clusters. The issues related to this algorithm are: the type of detected clusters; the separation of attribute values by their weights is non-intuitive and convergence depends on the number of basins. Each copy of the hypergraph contains two groups of attribute values, one with positive and another with negative weights, which define the two clusters

Zhang et al. [7] improved STIRR algorithm for guaranteed convergence. But it retained that the combination operator and the local modification operations be defined by the user based on concrete data. Also the detected clusters are affected by the post-processing required to generate the actual clusters from the basin weights upon reaching the fix point which is complex.

COOLCAT proposed by Barbara et al [8] is based on entropy reduction within the clusters. The initial cluster representatives are selected as a set of k tuples such that the minimum pairwise distance among them is maximized. The remaining tuples of the data set are assigned clusters such that the overall entropy of clusters is minimized.

Another approach based on cluster entropy measures was presented by Cristofor et al.[9] for categorical attributes. It uses genetic algorithms with crossover and mutation operators to heuristically improve the purity of the generated clusters. The drawback is that quality of the clusters depends on a-priori knowledge of the contribution of individual attributes towards the desired clustering.

Guha et al. presented ROCK (RObust Clustering using linKs)[10], a popular clustering algorithm for categorical attributes. It presents the concept of links between tuples which depend on similarity. The number of links thus indicates the number of records which are most similar to a record. Thus, this approach seems better than those which compare attribute values without considering their co-occurrence within a record. The objective function to be optimized is defined over the number of links in an agglomerative hierarchical fashion. Initially each tuple is assumed to be a separate cluster. Clusters are merged based on the closeness between clusters, defined by the number of links among the tuples of each. The major drawback is high complexity (cubic in the number of records), making it unsuitable for large datasets.

An overall summary information of the entire dataset is used in CACTUS [11], proposed by Ganti et al. It is based on combinatorial search. Unlike earlier algorithms it characterizes the detected categorical clusters through inter- and intra-attribute summaries. CACTUS first computes cluster projections onto the individual attributes. The authors assume the existence of a distinguishing number μ that represents the minimum size of the distinguishing sets which are attribute value sets that uniquely occur within only one cluster. The distinguishing sets are then extended to cluster projections. Finally, cluster projections can be combined to clusters candidates over multiple attributes which are validated against the original dataset.

Information of dataset required for clustering can be quantified using an Information Bottleneck (IB) framework, as suggested by Andritsos et al[12]. Their algorithm LIMBO[12] is a scalable hierarchical categorical clustering algorithm which can produce clusterings of different sizes in a single execution. A new distance measure for categorical attribute values is defined based on IB framework which can be used to cluster both tuples and values. The data model is memory bound, hence large data sets can be handled. It has been shown to be comparatively better than COOLCAT and ROCK.

Ahmad and Dey [13] attempted to alleviate the short-comings of Huang's cost function. The key differences are that Huang uses a binary-valued distance for categorical attributes, and all categorical attributes are weighed by a user-given parameter which controls the contribution of the categorical attributes to the distance function computed during the clustering process. While in Ahmad and Dey's method, the contribution of a categorical attribute is inherent in the distance measure itself and the user is not required to specify it. This contribution is a function of co-occurrence of values and thereby controls the grouping of similar elements that have similar values in a larger number of significant attributes. This distance measure can work well for mixed as well as pure numeric and categorical data sets. Results obtained over a number of mixed data sets using the proposed distance measure along with k-means clustering algorithm.

DILCA (DIstance Learning of Categorical Attributes) [14] a new method named to compute distances between values of a categorical variable and apply this technique to clustering categorical data by a hierarchical approach. This approach is independent from the specific clustering algorithm.

In 2013 Hassanein and Elmelegy [15] proposed two new concepts of similarity for categorical data, namely, the Standard Deviation of Standard Deviation Significance and Standard Deviation of Standard Deviation Dependence. The significance and dependence of attributes are concepts taken from rough set theory. Authors demonstrate the performance of the proposed algorithms compared with others; they are efficient and can handle uncertainty together with categorical data.

3. PROPOSED METHOD

For clustering categorical data, a pre-processing step is required to convert it into equivalent numeric data. Thereafter, a very simple and straightforward clustering approach is used.

3.1 Converting categorical values to numerical values

Certain categorical attributes are represented by ordinal numbers which leads to discrepancies when directly used in distance calculations. Also, the characteristic of any attribute space should also be reflected in its numeric equivalent or the similarity measure. We present a simple technique to convert all the non-numeric valued attributes to equivalent numeric attribute which when used in distance computation will reflect appropriate similarity or dissimilarity.

For a categorical attribute A, let the set of values that A can have be V= {v1, v2, v3,…… vl}. Then range (A) is defined as

Range (A) = max (V) –min (V)

Where max (V) is maximum value of A and min(V) is minimum value of A, if v1,v2,v3…. vn are number representation of values. If attribute A cannot be represented as numerical value, then

Range (A) =|V|= l

Frequency of occurrence of a particular value vi for A in a data set is used to compute its Prominence

Prominence (vi) = (number of times vi occurs in the dataset for attribute A)/n

The numerical equivalence of a categorical value is then calculated as

$$\text{Num-value}_i = v_i * \text{Prominence}(v_i)$$

Where n is the number of instances in the dataset.

Thus, this numerical equivalence when used for computing distance between data points will indicate two data points more similar if their categorical values have equal prominence.

3.2 Algorithm for Clustering

The basic idea is to capture the variation of each dimension separately. This is used to calculate a kind of dimension summary that can be used to assign a cluster label to any data point based only on that dimension. We begin with only first dimension and clusters are initialized according to it. Next, only two dimensions are considered for calculating distance from cluster representatives. Further, first three dimensions are considered. Thus, the distance calculations are different in every iteration and cluster labels are updated accordingly.

Let the dataset consist of n data points with m dimensions each. That is, every datapoint can be considered as a tuple of m values$\langle value_1, value_2, \dots, value_m \rangle$. Every categorical attribute is represented as explained above. The number of clusters is pre-decided and is input to the algorithm.

Step 1: Variation of each dimension is computed as, δi =

$$\frac{max_i - min_i}{k}, 1 \le i \le m$$

where max_i is maximum value if i[th] dimension and min_i is minimum value of i[th] dimension.

Step 2: Initial clusters are formed using following conditions, for any data point,

if $min_1 + j * \delta_1 \le value_1 < min_1 + (j+1) * \delta_1$then the data point belongs to cluster j.

Step 3: Centroid of each cluster is computed as mean of all cluster points

Step 4: For every secondary dimension, 2≤j≤m, repeat the following

Step 4.1: Reshape every cluster based on the condition: for every data point if $|value_{ij} - value_{cj}| > \delta_j$, then the data point is reconsidered based on dimension j. Here, $value_{ij}$ is value of value of the ith data point's jth dimension and $value_{cj}$ is value of the centroid's j[th] dimension.

Step 4.2: for each reconsidered data point, compute distance from each centroid up to j[th] dimension as $dist_c = \sum_{l=1}^{l=j} |value_{il} - value_{cl}|$. Decide the cluster of the data point according to the minimum distance.

4. IMPLEMENTATION RESULTS

We implement our algorithm using MATLAB. The performance is compared with other clustering algorithms over two metrics: cluster recovery and precision. Cluster recovery is measured per cluster as the ratio of the number of data points correctly assigned in cluster to the actual number of data points belonging to the cluster. This can be measured only for datasets where classes are known. Precision is an overall performance measure; the ratio of correctly clustered data points to total number of data points. A high precision is a direct measure of effectiveness of any clustering algorithm. It can be calculated as

$$r = \frac{\sum_{i=1}^{i=k} a_i}{n}$$

where a_i is the number of instances assigned correctly cluster number i, k is total number of clusters, and n is total number of instances(datapoints).

The proposed clustering algorithm has been tested on datasets available at [16]. Many researchers have used these datasets to demonstrate performance of their algorithms. Hence, they are good for comparison purpose in terms of cluster recovery and precision.

Mushroom dataset: It contains 8,124 tuples, each representing a mushroom characterized by 22 attributes, such as color, shape, odor, etc. The total number of distinct attribute values is 117. Each mushroom is classified as either poisonous or edible. There are 4,208 edible and 3,916 poisonous mushrooms in total. There are 2,480 missing values. The results of clustering over mushroom dataset using our algorithm are shown in Table 1.

Table 1. Performance of Proposed Algorithm over Mushroom dataset

	Edible	**Poisonous**
Indicated correctly by Proposed Algorithm	3960	3406
Ideal	4208	3916
Cluster Recovery	0.87	0.94
Precision	90.67%	

Vote dataset: It contains 435 tuples of votes from the U.S. Congressional Voting Record of 1984. Each tuple is a congress-person's vote on 16 issues and each vote is boolean, either YES or NO. Each congress-person is classified as either Republican or Democrat. There are a total of 168 Republicans and 267 Democrats. There are 288 missing values that we treat as separate values. The results of clustering over vote dataset using our algorithm are shown in Table 2.

Table 2. Performance of Proposed Algorithm over Vote dataset

	Republicans	**Democrats**
Indicated correctly by Proposed Algorithm	162	234
Ideal	168	267
Cluster Recovery	0.88	0.96
Precision	91.03%	

Heart Disease Dataset: This data generated at the Cleveland Clinic, is a mixed data set with eight categorical (one ordered, three binary and four nominal) and five numeric features. It contains 303 instances belonging to two classes – normal (164) and heart patient (139). The results of clustering over the Cleveland heart disease dataset are shown in Table 3.

Table 3. Performance of Proposed Algorithm over Heart Disease dataset

	Normal	**Heart Patient**
Indicated by Proposed Algorithm	139	136
Ideal	164	139
Cluster Recovery	0.85	0.98
Precision	90.75%	

The scalability of the proposed algorithm can be seen in in Fig 1, the results prove that the run-time of our proposed algorithm is almost linear in terms of number of instances. It shows the runtime of algorithm over a dataset of 20 attributes and four clusters, varying with increasing number of instances.

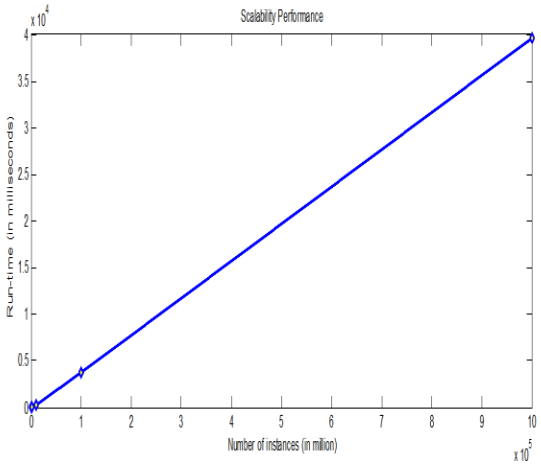

Fig 1: Plot of Runtime of proposed algorithm against increasing number of instances

The scalability in terms of dimensions can be understood from Fig 2 the graph of runtime against increasing number of dimensions. At fixed value of k=2, and number of data points 100, number of dimensions is increased.

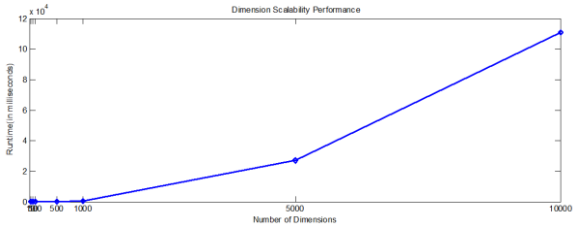

Fig 2: Plot of Runtime of proposed algorithm against increasing number of dimensions

Effect of number of clusters on the runtime should be studied because the number of centroids depends on number of clusters, hence the time required for distance calculation increases when the number of clusters increase. Fig 3 shows the variation in run-time with increasing number of clusters over dataset of 1000 instances of 20 dimensions. It shows that the runtime of the proposed algorithm is linear in number of clusters.

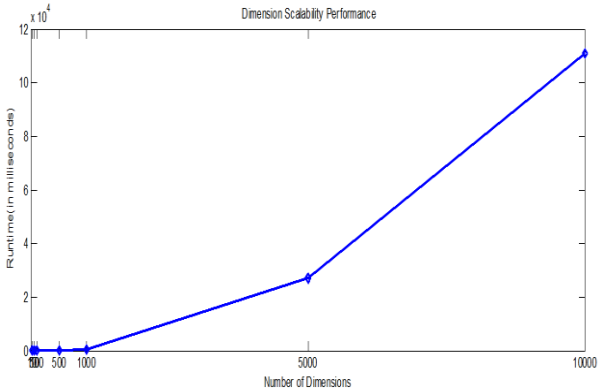

Fig 3: Plot of Runtime of proposed algorithm against increasing number of clusters

The proposed algorithm can be compared on the basis of accuracy (precision) obtained over the publicly available datasets with other clustering algorithms for which results on same datasets have been published in literature. The algorithms proposed by Wu et al [17], Cao et al [18], Khan and Ahmad [19] are all variants of k-modes. In [14], k-modes have been combined with a new dissimilarity measure DILCA and similar combination has been done with hierarchical clustering. Table 5 compares these all with our proposed algorithm.

Table 5. Comparison of Proposed Method with Other Clustering Algorithms According to Clustering of Mushroom Dataset

	k-means	**Wu**	**Cao**	**Khan and Ahmad**	**K-modes DILCA**	**HCL-DILCA**	**Proposed Algorithm**
Precision over Mushroom dataset	0.3762	0.8754	0.8754	0.8815	0.8902	0.8902	0.9067

Table 6 compares the results of our algorithm over Congressional vote dataset with those published in [13]. ROCK [10] is a popular technique for categorical clustering, Huang's method [5] is based on k-modes and technique of Ahmad and Dey [13] is a variation of k-means. OCIL [20] is a recently proposed categorical clustering algorithm which does not use a priori knowledge of number of clusters.

Table 6 Comparison of Proposed Method with Other Clustering Algorithms According to Clustering of Vote Dataset

	ROCK	Huang	Khan and Ahmad	Ahmad and Dey	K-modes DILCA	HCL-DILCA	OCIL	Proposed Algorithm
Precision over Vote data set	0.7931	0.8367	0.8506	0.8667	0.8759	0.8959	0.8787	0.9103

Table 7 compares the results of our algorithm over Heart Disease dataset with those published in [13]. The popular algorithms for mixed data have been selected for comparison – SBAC [21], Huang[22], ECOWEB[23]and Ahmad-Dey [13]. OCIL [20] is a recently proposed categorical clustering algorithm which does not use a priori knowledge of number of clusters.

Table 7 Comparison of Proposed Method with Other Clustering Algorithms According to Clustering of Heart Disease Dataset

	SBAC	Huang	ECOWEB	Ahmad and Dey	OCIL	Proposed Algorithm
Precision over Heart Disease dataset	0.7525	0.6667	0.74	0.8482	0.8313	0.9076

Thus, it can be concluded that the clustering performance of the proposed algorithm is better than many popular algorithms.

5. CONCLUSION

Many distance measures for categorical attributes have been suggested by researchers, but they need to modify the clustering algorithm accordingly. We present a technique to convert categorical attribute values to equivalent numeric values such that distance measures of conventional clustering algorithms can also be used. Moreover, we also presented a simple clustering method that is computationally light yet accurate and scalable, both in terms of number of instances and attributes.

6. REFERENCES

[1] E.W. Forgy (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics 21: 768–769.

[2] J.A. Hartigan (1975). Clustering algorithms. John Wiley & Sons, Inc.

[3] Hartigan, J. A.; Wong, M. A. (1979). "Algorithm AS 136: A K-Means Clustering Algorithm". Journal of the Royal Statistical Society, Series C 28 (1): 100–108. JSTOR 2346830.

[4] M. J. Zaki and M. Peters, "Click: Mining subspace clusters in categorical data via k-partite maximal cliques," in Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on. IEEE, 2005, pp. 355–356.

[5] 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.

[6] GIBSON, D., KLEINBERG, J., and RAGHAVAN, P. 1998. Clustering categorical data: An approach based on dynamic systems. In Proceedings of the 24th International Conference on Very Large Databases, 311-323, New York, NY.

[7] Y. Zhang, A. Fu, C. Cai, and P. Heng. Clustering categorical data. In Proceedings of the ICDE, page 305, 2000.

[8] D. Barbar´a, Y. Li, and J. Couto, "Coolcat: an entropy-based algorithm for categorical clustering," in Proceedings of the eleventh international conference on Information and knowledge management. ACM, 2002, pp. 582–589.

[9] D. Cristofor and D. Simovici. An information-theoretical approach to clustering categorical databases using genetic algorithms. In 2nd SIAM ICDM, Workshop on clustering high dimensional data, 2002.

[10] Guha, S., Rastogi, R., & Shim, K. (1999). Rock: a robust clustering algorithm for categorical attributes. In Proceedings of the 15th international conference on data engineering, 23–26 March 1999, Sydney, Austrialia (pp. 512–521). IEEE Computer Society.

[11] GANTI, V., GEHRKE, J. and RAMAKRISHNAN, R. 1999a. CACTUS-Clustering Categorical Data Using Summaries. In Proceedings of the 5th ACM SIGKDD, 73-83, San Diego, CA.

[12] P. Andritsos, P. Tsaparas, R. J. Miller, and K. C. Sevick, "Limbo: Scalable clustering of categorical data," in Advances in Database Technology-EDBT 2004. Springer, 2004, pp. 123–146.

[13] Ahmad, A., & Dey, L. (2011). A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. Pattern Recognition Letters, 32, 1062–1069.

[14] D. Ienco, Ruggero G. Pensa and R. Meo, "Context-Based Distance Learning for Categorical Data Clustering," Advances in Intelligent Data Analysis VIII Lecture Notes in Computer Science Volume 5772, 2009, pp 83-94.

[15] W. A. Hassanein and Amr A. Elmelegy, "clustering algorithms for Categorical data using concepts of Significance and dependence of Attributes", European Scientific Journal January 2014 edition vol.10, No.3, pp 381-400.

[16] UCI Machine Learning Repository, http://ics.uci.edu/ mlearn/MLRepository.html

[17] Wu, S., Jiang, Q., & Huang, J. Z. (2007). A new initialization method for clustering categorical data. In Proceedings of the 11th Pacific-Asia conference on advances in knowledge discovery and data mining PAKDD'07 (pp. 972–980). Berlin, Heidelberg: Springer-Verlag.

[18] Cao, F., Liang, J., & Bai, L. (2009). A new initialization method for categorical data clustering. Expert Systems and Applications, 36, 10223–10228.

[19] S S Khan and A Ahmad, "Cluster center initialization algorithm for K-modes clustering", Expert Systems with Applications 40 (2013) 7444–7456.

[20] Y Cheung and H Jia, "Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number", Pattern Recognition 46 (2013) 2228–2238. Available at http://dx.doi.org/10.1016/j.patcog.2013.01.027

[21] C. Li, G. Biswas, Unsupervised learning with mixed numeric and nominal data, IEEE Transactions on Knowledge and Data Engineering 14 (4) (2002) 673–690.

[22] J.Z. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in k-mean type clustering, IEEE Transactions on PAMI 27 (5) (2005).

[23] Y. Reich, S.J. Fenves, The formation and use of abstract concepts in design, in: D.H. Fisher, M.J. Pazzani, P. Langley (Eds.), Concept Formation: Knowledge and Experience in Unsupervised Learning, Morgan Kaufman, Los Altos, Calif, 1991, pp. 323–352.