



HAL
open science

Cube Sampled K-Prototype Clustering for Featured Data

Seemandhar Jain, Aditya A Shastri, Kapil Ahuja, Yann Busnel, Navneet Pratap Singh

► **To cite this version:**

Seemandhar Jain, Aditya A Shastri, Kapil Ahuja, Yann Busnel, Navneet Pratap Singh. Cube Sampled K-Prototype Clustering for Featured Data. INDICON 2021: IEEE 18th India Council International Conference, Dec 2021, Guwahati, India. pp.1-6, 10.1109/INDICON52576.2021.9691727 . hal-03515281

HAL Id: hal-03515281

<https://imt-atlantique.hal.science/hal-03515281>

Submitted on 6 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cube Sampled K -Prototype Clustering for Featured Data

Seemandhar Jain

Computer Science and Engineering
Indian Institute of Technology Indore
Indore, India
cse170001046@iiti.ac.in

Aditya A. Shastri

Information Technology
NMIMS, Shirpur
Shirpur, India
aditya.shastri@nmims.edu

Kapil Ahuja*

Math of Data Science & Simulation (MODSS) Lab
Indian Institute of Technology Indore
Indore, India
kahuja@iiti.ac.in

Yann Busnel

Network Systems, Cybersecurity and Digital Law Department
Institut Mines-Telecom Atlantique
Rennes, France
yann.busnel@imt-atlantique.fr

Navneet Pratap Singh

Computer Science and Engineering
NMIMS Indore
Indore, India
navneet.diat@gmail.com

Abstract—Clustering large amount of data is becoming increasingly important in the current times. Due to the large sizes of data, clustering algorithm often take too much time. Sampling this data before clustering is commonly used to reduce this time. In this work, we propose a probabilistic sampling technique called cube sampling along with K -Prototype clustering. Cube sampling is used because of its accurate sample selection. K -Prototype is most frequently used clustering algorithm when the data is numerical as well as categorical (very common in today's time). The novelty of this work is in obtaining the crucial inclusion probabilities for cube sampling using Principal Component Analysis (PCA).

Experiments on multiple datasets from the UCI repository demonstrate that cube sampled K -Prototype algorithm gives the best clustering accuracy among similarly sampled other popular clustering algorithms (K -Means, Hierarchical Clustering (HC), Spectral Clustering (SC)). When compared with unsampled K -Prototype, K -Means, HC and SC, it still has the best accuracy with the added advantage of reduced computational complexity (due to reduced data size).

Index Terms—Sampling, Cube Sampling, Clustering, K -Prototype Clustering, Principal Component Analysis, Clustering Accuracy.

I. INTRODUCTION

Due to the pervasiveness of social media and the subsequent transformation of the world digitally, an extensive amount of data is being generated. According to a survey [1], more than 2.5 quintillion bytes of data is produced every day, which is increasing exponentially. Since not all of this data is useful, it is important to obtain some meaning from it to help in decision making.

Clustering is the technique to group together the data items having similar behavior. It is often used to get the initial perception of data and is widely used. Some of the popular clustering algorithms include K -Means, K -Modes, K -

Prototype, Hierarchical Clustering (HC), Spectral Clustering (SC), Partitioning Around Medoids (PAM), Expectation Maximization (EM) Clustering, Affinity Propagation Clustering etc. [2]–[7]. Each of these has its limitations [6]. K -Prototype is a commonly used clustering algorithm because of its ability to handle numerical and categorical data together [8]. To achieve this, it uses a combination of the K -Means and the K -Modes algorithms and is easy to implement as well.

Clustering becomes difficult when the size of the data is very large. One of the ways to develop an efficient clustering algorithm is to sample this data. There are several sampling techniques that are commonly used. This include random sampling, stratified sampling, Vector Quantization (VQ), pivotal sampling, cube sampling etc. [9]–[12]. We focus on cube sampling because *firstly* it belongs to the class of probabilistic sampling that are accurate. *Secondly*, it is independent of the order of the data selected to compute the crucial inclusion probabilities [12]. To sum up, we propose cube sampled K -Prototype clustering algorithm with specific focus on data that contains numerical and categorical features.

There have been a few attempts to perform clustering while using a sampling technique to reduce the data [10], [11]. In [10], authors have proposed use of VQ sampling technique along with SC. VQ is a very basic sampling technique with comparatively low accuracy. Moreover, the algorithm in [10] is specifically designed for genome data. Another work is [11], where a different probabilistic sampling technique called pivotal sampling is used with SC. Although, pivotal sampling is good in terms of accuracy, this algorithm is tested on only plant phenotypic data. This paper also uses pivotal sampling with HC. However, it also has same limitations as pivotal sampling with SC. In another work [13], the authors have used random sampling with the ensemble clustering algorithm. However, they do not show the efficacy of their approach on large datasets.

*Corresponding author
978-1-6654-4175-9/21/\$31.00 ©2021 IEEE

As evident from above discussion, cube sampling has many advantages and it has not been coupled with any of the commonly used clustering algorithms, which is the focus of this work. As mentioned above, besides combining cube sampling with K -Prototype, we integrate it with K -Means, HC, and SC as well. The most novel contribution of this work is in computing the crucial inclusion probabilities for the cube sampling technique using the concept of Principal Component Analysis (PCA).

The proposed algorithm is evaluated with extensive experiments that are performed on six labeled datasets from UCI repository with varying sizes. Clustering Accuracy is used as a metric for comparison. We perform four sets of experiments. *First*, we compare four commonly used clustering algorithms (i.e. unsampled K -Prototype, K -Means [2], HC [4], and SC [5]), where the K -Prototype gives the best accuracy. *Second*, when using standard random sampling [9], we again compare these algorithms. Here, again, K -Prototype with random sampling gives the best accuracy. *Third*, we perform same comparison with our cube sampling. *Fourth* and *finally*, we compare our cube sampled K -Prototype with unsampled other clustering algorithms mentioned above. Results for both the third and the fourth sets of comparisons show that our proposed algorithm performs the best. Also, our cube sampled K -Prototype has the added advantage of reduced computational complexity (since the input data is reduced by cube sampling before clustering).

The rest of this paper is organized as follows. Section II provides a detailed discussion of our proposed algorithm. Experimental setup is presented in Section III. Section IV gives the experimental results. Finally, conclusions and future work are provided in Section V.

II. PROPOSED ALGORITHM

In this section, we first explain the concept of cube sampling including our novel way of obtaining the inclusion probabilities. Subsequently, we discuss the K -Prototype clustering algorithm, followed by our cube sampled K -Prototype clustering.

A. Application of Cube Sampling

Cube sampling is a type of probabilistic sampling technique that selects a balanced sample [12]. Consider a population U of size n such that $U = \{x_1, x_2, \dots, x_n\}$, where $x_i \in \mathbb{R}^m$ for $i = 1, 2, \dots, n$ and m is the total number of features. A sample S of size N is said to be balanced if the following equality is satisfied [14].

$$\sum_{j \in S} \frac{x_j}{\pi_j} = \sum_{i \in U} x_i, \quad (1)$$

where π_j is the inclusion probability of the j^{th} unit and $j = 1, 2, \dots, N$. Obtaining the inclusion probabilities of all the units in the population, denoted by π_i , forms an important aspect of this probability based sampling technique. These probabilities are not sample invariant, however, they do not affect the working of cube sampling.

Cube sampling obtains the samples in two phases; the flight phase and the landing phase. The flight phase starts by obtaining the initial values of π_i for $i = 1, 2, \dots, n$ (our unique contribution; discussed in the subsequent paragraphs). These initial probabilities are updated in each iteration of flight phase using a random walk. At each iteration, at least one unit is either included or excluded from the sample. At the end of this phase, inclusion probabilities of all the units are either equal to 0 (eliminated from the sample) or equal to 1 (included in the sample).

After the flight phase, if the balancing equation is not exactly satisfied or, inclusion probability of even one unit is neither 0 nor 1, then there is a need for the landing phase. The goal of the landing phase is to find a sample S such that it is almost balanced (i.e. as close to balanced as possible).

Let p be the number of units having inclusion probabilities neither 0 nor 1. Here, we obtain all the possible samples by taking the probability values of these p units both 0 and 1. As there are two possible values for each of these p units, we get a total of 2^p possible samples. Finally, we select the sample that closely satisfies the balancing condition.

Next, the working of cube sampling is explained using an example of a random population of size six. Let the initial probabilities values are $\pi^0 = (0.5, 0.5, 0.5, 0.5, 0.5, 0.5)$, and we need to select three samples. The cube sampling transforms these initial inclusion probabilities as shown in (2). Upon convergence, at the j^{th} step, units with the final probability as 0 are discarded and units with the final probability as 1 are selected in the sample.

$$\begin{aligned} \pi^0 = \begin{pmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 0.5 \end{pmatrix} &\rightarrow \begin{pmatrix} 0.6 \\ 0.6 \\ 0 \\ 0.6 \\ 0.6 \\ 0.6 \end{pmatrix} \rightarrow \begin{pmatrix} 0.6 \\ 0 \\ 0 \\ 1 \\ 0.8 \\ 0.6 \end{pmatrix} \\ &\rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0.5 \\ 0.5 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \pi^{j^{\text{th}}} \quad (2) \end{aligned}$$

Mathematically, the initial inclusion probabilities (π^0) are updated using the below equation [14].

$$\pi^{\ell+1} = \begin{cases} \pi^\ell + \lambda_1^*(\ell)\mathbf{u}(\ell) & \text{with probability } \mathbf{q}_1(\ell) \\ \pi^\ell - \lambda_2^*(\ell)\mathbf{u}(\ell) & \text{with probability } \mathbf{q}_2(\ell) \end{cases} \quad (3)$$

Where

- $\mathbf{q}_1(\ell) = \lambda_2^*(\ell) / \{\lambda_1^*(\ell) + \lambda_2^*(\ell)\}$,
- $\mathbf{q}_2(\ell) = 1 - \mathbf{q}_1(\ell)$,
- $\mathbf{u}(\ell)$ is random kernel of dataset U ,
- $\lambda_1^*(\ell)$ is the largest value satisfying $0 \leq \pi^\ell + \lambda_1(\ell)\mathbf{u}(\ell) \leq 1$, and

- $\lambda_2^*(\ell)$ is the largest value satisfying $0 \leq \pi^\ell - \lambda_2(\ell)\mathbf{u}(\ell) \leq 1$.

Inclusion probability of an unit x_i is defined as the probability with which the unit is selected in the sample S and, as earlier, is denoted by π_i . Here, we propose a novel and an efficient way to calculate the initial inclusion probabilities using Principle Component Analysis (PCA) [15]. Although PCA has been used earlier to sample data, it has never been combined with any probabilistic sampling techniques. PCA transforms the data by projecting it onto a single axis where elements are maximally deviated. The process of calculating the inclusion probabilities is described in the following equations [12]:

$$\pi_i = \min \left[1, h^{-1}(n) \frac{x_i}{X} \right], i \in U \quad (4)$$

where, as earlier, U is the total population and h is defined as

$$h(z) = \sum_{i \in U} \min \left(z \frac{x_i}{X}, 1 \right), \quad (5)$$

with $\sum_{i \in U} \pi_i = N$ and $X = \sum_{i \in U} x_i$.

B. Our Algorithm

The K -Prototype algorithm is an integration of the K -Means algorithm and the K -Modes algorithm to handle mixed data types. Here, the basic idea is similar to K -Means with a different distance criteria to handle numerical and categorical features. The aim is to minimize the cost function by partitioning the dataset into k clusters. This cost function is given as

$$C = \sum_{l=1}^k \sum_{j=1}^N p_{jl} \text{DIST}(x_j, q_l), \quad (6)$$

where $p_{jl} \in \{0, 1\}$ denotes the membership of data point x_j in cluster l , q_l denotes the cluster center for cluster l and $\text{DIST}(x_j, q_l)$ is the distance criteria (dissimilarity measure), which is given as

$$\text{DIST}(x_j, q_l) = \sum_{r=1}^{m_r} (x_{jr} - q_{lr})^2 + \gamma_l \sum_{t=1}^{m_t} \delta(x_{jt}, q_{lt}), \quad (7)$$

where,

- r is index for numerical features and m_r is the total number of numerical features for each data point x_j ,
- x_{ir} denotes the value of the r^{th} numerical feature of x_j ,
- q_{lr} is the mean of the r^{th} numerical feature in the cluster l ,
- γ_l is the weight factor for categorical features for cluster l ,
- t is index for categorical features and m_t is the total number of categorical features of x_j ,
- x_{jt} denotes the t^{th} categorical feature of x_j ,
- q_{lt} is the mode of the t^{th} categorical feature in the cluster l , and

- $\delta(a, b) = 1$ for $a \neq b$ and $\delta(a, b) = 0$ for $a = b$.

The proposed algorithm (Cube sampled K -Prototype algorithm) is given in Algorithm 1 and the broad framework is graphically represented in Fig. 1.

Algorithm 1: Proposed Algorithm

Input: A - Data Matrix ($n \times m$ rows), N - Sampling size, k - Number of clusters

Output: Clusters

- 1 π denotes initial probabilities that is computed as discussed in Section II-A
 - 2 π_n denotes final probabilities that is obtained after the flight and the landing phase of cube sampling (also discussed in Section II-A)
 - 3 Generate sample S from π_n by using units with probability 1
 - 4 Choose k cluster centers from S randomly to form q_{old}
 - 5 Find the distance between elements and cluster centers as described in (7) to form q .
 - 6 **while** $q_{old} \neq q$ **do**
 - 7 $q_{old} = q$
 - 8 Update q using (7) again
 - 9 Reverse map the remaining points based on their similarity with all clusters.
 - 10 **return** $ClusterDetails$
-

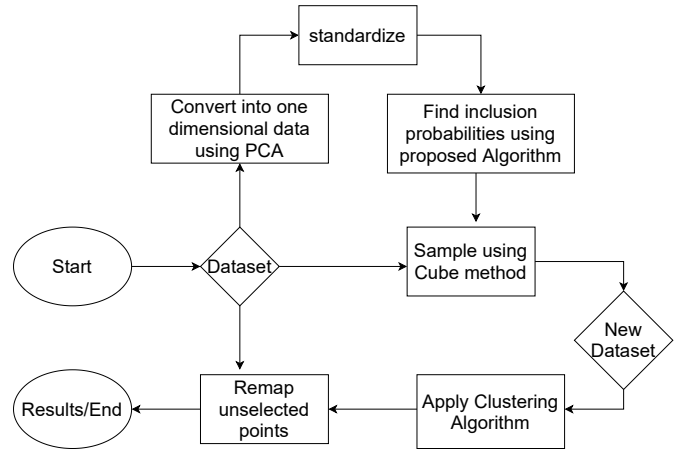


Fig. 1: Framework for the proposed algorithm.

Next, we perform the complexity analysis of our proposed approach.

- 1) Calculating inclusion probabilities: $\mathcal{O}(m^2n + m^3)$
- 2) Cube sampling: $\mathcal{O}(n)$
- 3) K -Prototype algorithm: $\mathcal{O}(kNm)$
- 4) Reverse mapping: $\mathcal{O}((n - N)k)$

Here, n is the number of data points in the original dataset, N is the sampling size, m denotes the number of features, and k is the number of clusters. Thus, the total time complexity of our algorithm is $\mathcal{O}(n(k + m^2) + kNm + m^3)$.

III. EXPERIMENTAL SETUP

In this section, we first briefly discuss the datasets used for our experiments and then describe the criteria used to check the goodness of our proposed algorithm. For experiments, we use six datasets with varying sizes. These include labeled datasets downloaded from the UCI repository [16]. Table I provides a brief summary of these datasets, where all the columns are self explanatory.

Since we use labeled datasets for our experiments, we check the quality of our clustering by using Clustering Accuracy (CA), which is calculated as given below [3].

$$CA = \sum_{l=1}^k \frac{\max(C_l | L_l)}{|U|}, \quad (8)$$

where C_l is the set of elements in the l^{th} cluster, L_l is the set of all the class labels in the l^{th} cluster, $\max(C_l | L_l)$ is the number of elements with the majority label in the l^{th} cluster, and U is the population size.

The experiments are performed on a machine with Intel® Core™ i7-7500U CPU 2.90 GHz and 16 GB RAM. Python 3.7 is used to implement all the algorithms.

IV. RESULTS

In this section, we show the effectiveness of our proposed algorithm (i.e. Cube sampled K -Prototype). For this, we do four various comparisons here.

- Without performing any sampling, we first compare K -Prototype clustering with other commonly used clustering techniques (K -Means, Hierarchical Clustering (HC), Spectral Clustering (SC)).
- When using frequently used random sampling, we again compare K -Prototype with same set of clustering algorithms.
- Next, when using our proposed cube sampling, we further compare all the above-mentioned clustering algorithms.
- Finally, we compare our cube sampled K -Prototype algorithm with unsampled other clustering algorithms mentioned above.

Table II compares four different clustering algorithms (K -Prototype, K -Means, Hierarchical Clustering (HC), and Spectral Clustering (SC)) without applying any prior sampling of data. From this table, it is clear that CA values for K -Prototype are greater than or equal to the other algorithms for all the datasets. Thus, K -Prototype performs best among all.

Next, in Table III, we again compare the four clustering algorithms mentioned above but on sampled data (using random sampling). Without loss of generality, we take sample size as 100 for the two relatively smaller datasets (German and Heart), while sample size is taken as 1000 for the other datasets. From this table, we again observe that K -Prototype with random sampling performs best when compared with all other algorithms with random sampling. Use of random

sampling deteriorate the CA values substantially, which is not the case with our sampling technique (discussed below).

Further, in Table IV, we perform the same set of comparisons as in Table III except using data obtained after applying cube sampling. The sample sizes here are same as in the previous set of comparisons. We can observe from this table that the CA values for K -Prototype with cube sampling (our proposed algorithm) is greater than all the other algorithms with cube sampling.

Finally, we compare our proposed algorithm (cube sampled K -Prototype) with the unsampled four clusterings as discussed before. For this, we pick the data from the relevant columns of Table II and Table IV. This set of comparisons is given in Table V below. We observe from this table that our algorithm performs better than all others.

To demonstrate that our technique gives good result irrespective of the size of the sample, in Fig. 2, we plot the CA values of our proposed algorithm for different sample sizes and different datasets. On the x -axis, sample size is given, and on the y -axis, CA values are given. From these graphs, it is clear that sample size does not affect the CA values. Thus, use of cube sampling does not deteriorate the quality of clustering as done by K -Prototype. It also has an added advantage of reduced computational complexity (as it works on reduced dataset making it faster).

V. CONCLUSION

In this paper, we propose a cube sampled K -Prototype algorithm for grouping large data. The most innovative aspect of our algorithm is computation of inclusion probabilities in cube sampling using PCA. Experiments on labeled UCI datasets with varying sizes show that our proposed algorithm gives best clustering accuracy when compared with other commonly used unsampled clustering algorithms (K -Means, HC, and SC). Also, use of cube sampling does not deteriorate the accuracy of the K -Prototype algorithm. Furthermore, our algorithm also has the added advantage of reduced computational complexity (since the size of the data is reduced by cube sampling). In future, we plan to adapt our algorithm for datasets with millions of units. We also aim to combine cube sampling with other accurate clustering algorithms. We intend to adapt our algorithm for Hadoop framework to make it parallel as well [17].

REFERENCES

- [1] J. Milenkovic, "30 Eye-opening big data statistics for 2020: patterns are everywhere," <https://kommandotech.com/statistics/big-data-statistics/>, Accessed: June 2020.
- [2] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.
- [3] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Zomaya, et al., "A survey of clustering algorithms for big data: taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, pp. 267–279, 2014.
- [4] P. Painkra, R. Shrivatava, S. Nag and N. Kumar, "Clustering analysis of soybean germplasm (Glycine max L. merrill)," *The Pharma Innovation Journal*, vol. 7, pp. 781–786, 2018.
- [5] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

TABLE I: Dataset information from the UCI repository

Dataset	# of records	# of features	# of classes	Dataset Descriptions
German	1000	20	2	Categorizes individuals based on a set of factors as either good or bad credit risks.
Heart	303	13	2	Classifies if a person has heart disease or not.
Satellite	6435	36	2	Predicts the classification of neighborhood using multi-spectral values of pixels in a satellite picture.
Adult	45222	14	2	Classifies whether an individual earns more than 50k per year or not based on census data.
Shuttle	43500	9	7	Indicates whether or not a NASA Space Shuttle component will fail.
Kddcup99	24701	41	18	Simulates a range of intrusions in a military network environment, distinguishing between types of intrusions and standard connections (using a set of auditable data).

TABLE II: CA values for four different clustering algorithms on the six labeled datasets without prior application of sampling.

Dataset	<i>K</i> -Prototype	<i>K</i> -Means	HC	SC
German	0.68	0.53	0.62	0.66
Heart	0.59	0.52	0.53	0.52
Satellite	0.79	0.71	0.72	0.75
Adult	0.69	0.61	0.67	0.68
Shuttle	0.82	0.82	0.82	0.82
Kddcup99	0.69	0.59	0.69	0.68

TABLE III: CA values for four different clustering algorithms on the six labeled datasets with prior application of random sampling.

Dataset	N	<i>K</i> -Proto.	<i>K</i> -Means	HC	SC
German	100	0.51	0.39	0.43	0.45
Heart	100	0.48	0.41	0.39	0.42
Satellite	1000	0.63	0.53	0.55	0.56
Adult	1000	0.60	0.54	0.55	0.57
Shuttle	1000	0.69	0.68	0.68	0.69
Kddcup99	1000	0.41	0.37	0.39	0.40

TABLE IV: CA values for four different clustering algorithms on the six labeled datasets with cube sampling.

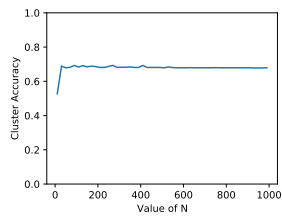
Dataset	N	<i>K</i> -Proto.	<i>K</i> -Means	HC	SC
German	100	0.68	0.52	0.60	0.64
Heart	100	0.59	0.50	0.52	0.51
Satellite	1000	0.78	0.69	0.70	0.73
Adult	1000	0.70	0.60	0.64	0.67
Shuttle	1000	0.82	0.80	0.81	0.82
Kddcup99	1000	0.69	0.56	0.66	0.67

TABLE V: CA values for our cube sampled *K*-Prototype algorithm and other four unsampled clustering algorithms on the six labeled datasets.

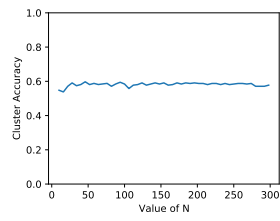
Dataset	N	Our Algo.	<i>K</i> -Proto.	<i>K</i> -Means	HC	SC
German	100	0.68	0.68	0.53	0.62	0.66
Heart	100	0.59	0.59	0.52	0.53	0.52
Satellite	1000	0.78	0.79	0.71	0.72	0.75
Adult	1000	0.70	0.69	0.61	0.67	0.68
Shuttle	1000	0.82	0.82	0.82	0.83	0.82
Kddcup99	1000	0.69	0.69	0.59	0.69	0.68

[6] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015.
 [7] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, 1996.
 [8] Z. Huang, "Extensions to the k-means algorithm for clustering large

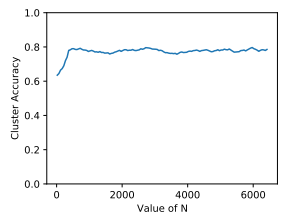
data sets with categorical values," *Data Mining and Knowledge Discovery*, vol. 2, pp. 283–304, 1998.
 [9] F. Olken and D. Rotem, "Simple random sampling from relational databases," University of California, 1986.
 [10] A. Shastri, K. Ahuja, M. Ratnaparkhe, A. Shah, A. Gagrani and A. Lal, "Vector quantized spectral clustering applied to whole genome sequences of plants," *Evolutionary Bioinformatics*, vol. 15, pp. 1-7, 2019.
 [11] A. Shastri, K. Ahuja, M. Ratnaparkhe and Y. Busnel, "Probabilistically sampled and spectrally clustered plant species using phenotypic characteristics," *PeerJ*, in press, 2021.
 [12] Y. Tillé, *Sampling Algorithms*, Springer, 2006.
 [13] F. Yang, X. Li, Q. Li and T. Li, "Exploring the diversity in cluster ensemble generation: random sampling and random projection," *Expert Systems with Applications*, vol. 41, pp. 4844-4866, 2014.
 [14] G. Chauvet and Y. Tillé, "A fast algorithm for balanced sampling," *Computational Statistics*, vol. 21, pp. 53–62, 2006.
 [15] S. Karamizadeh, S. Abdullah, A. Manaf, M. Zamani and A. Hooman, "An overview of principal component analysis," *Journal of Signal and Information Processing*, vol. 4, pp. 173, 2013.
 [16] D. Dua and C. Graff, "UCI machine learning repository," <http://archive.ics.uci.edu/ml>, Accessed: February 2021.
 [17] V. Nemade, A. Shastri, K. Ahuja and A. Tiwari, "Scaled and projected spectral clustering with vector quantization for handling big data", *Proc. of the 9th Symposium Series on Computational Intelligence (SSCI)*, IEEE, Bengaluru, India, pp. 2174–2179, 2018.



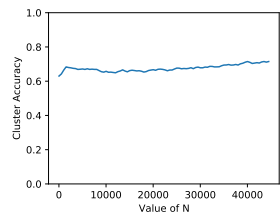
(a) German Dataset



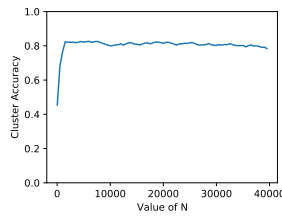
(b) Heart Dataset



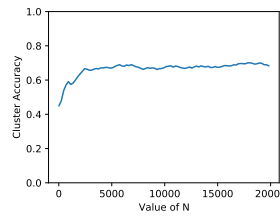
(c) Satellite Dataset



(d) Adult Dataset



(e) Shuttle Dataset



(f) Kddcup99 Dataset

Fig. 2: CA values of our cube sampled K -Prototype algorithm for different sample sizes and different datasets.