Empirical Comparative Analysis of 1-of-K Coding and K-Prototypes in Categorical Clustering

Fei Wang[1], Hector Franco[1], John Pugh[2], and Robert Ross[1]

[1] School of Computing, Dublin Institute of Technology, Ireland
[2] Nathean Technologies Ltd. Dublin, Ireland.
d13122837@mydit.ie

Abstract. Clustering is a fundamental machine learning application, which partitions data into homogeneous groups. K-means and its variants are the most widely used class of clustering algorithms today. However, the original k-means algorithm can only be applied to numeric data. For categorical data, the data has to be converted into numeric data through 1-of-K coding which itself causes many problems. K-prototypes, another clustering algorithm that originates from the k-means algorithm, can handle categorical data by adopting a different notion of distance. In this paper, we systematically compare these two methods through an experimental analysis. Our analysis shows that K-prototypes is more suited when the dataset is large-scaled, while the performance of k-means with 1-of-K coding is more stable. We believe these are useful heuristics for clustering methods working with highly categorical data.

Keywords: clustering, categorical data, k-means, k-prototypes, efficiency, clustering validity

1 Introduction

Clustering is a fundamental machine learning operation to partition data into homogeneous groups [15]. Different from classification, clustering looks at the intrinsic characteristics of data, rather than the relationship of the data with external labels. Identified data clusters should be "externally isolated and internally cohesive, implying a certain degree of homogeneity within clusters and heterogeneity between clusters" [26]. In other words, clustering aims to partition a set of objects into clusters such that the objects in the same cluster are more similar to each other than the objects in different clusters [15, 24].

Clustering has historically been the most popular of the unsupervised machine learning techniques. Typical applications include: (a) the discovery of underlying structure in data; (b) the classification of data based on its intrinsic nature; and (c) the compression of data [17]. As a fundamental method in data mining and machine learning, clustering has been applied a variety of fields, such as image segmentation, documents analysis, customer segmentation, workforce management, genome research in biology and so on [17]. It is also noted as an important part of unsupervised learning in most data mining and machine learning text books [21, 18, 6, 8, 29].

Clustering algorithms can be divided into four categories [29]: (a) Representative-based clustering, e.g. k-means; (b) Hierarchical clustering, e.g. agglomerative hierarchical clustering; (c) Density-based clustering, e.g. DBSCAN; and (d) Spectral and graph clustering, e.g. spectral clustering. Jain provides a useful detailed introduction to the progress and development of different kinds of clustering algorithms [17]. Given the multitude of clustering techniques, the primary question that needs to be answered for a given application is which algorithm should be chosen for a specific case. However, many see the answer to this question as being as complex as the range of algorithms themselves available. For a start, answering this question depends on the characteristics of source data, algorithms and the targets of clustering. As stated in [17, 28], no clustering algorithm can be universally used to solve all problems. Algorithms are always designed with some assumptions or restrictions, in this sense, it is important to have a clear idea about the conditions of the clustering. Even though lots of work can be implemented before the clustering to select algorithms, it is still often impossible to find the "best" algorithm, because the combinations of algorithms and conditions lead to a vast amount of work. Usually, a systematic comparison of some widely used algorithms is the pragmatic way to find out which algorithm to use.

Our own interest in clustering stems from its importance in customer segmentation. In commercial Business Intelligence applications, the ability to cluster data is a vital tool in order to provide insights into business data. For end users the most beneficial form of clustering is where little a priori knowledge such as the likely number of clusters is needed in advance of the commencement of the clustering process. We see the automatic parameterisation and execution of clustering processes as a goal for our work both from an academic and commercial perspective. We are particularly concerned with the problem of clustering data that has a high proportion of categorical data. Clustering highly categorical data has its own associated challenges but is yet of very real interest to a range of application types. We also pay much attention to the efficiency of the algorithm implementation, which is always a vital aspect for commercial applications.

In this paper, we outline an empirical comparison of k-means clustering with its derivative algorithm k-prototypes in the context of categorical data analysis. We begin in Section 2 with a brief recap of key issues in clustering for data with a high proportion of categorical data and introduce the two algorithms which we are focusing on. Then in Section 3 we outline the design for our empirical comparison of the algorithms in question. Section 4 presents the study results, before in Section 5 we draw conclusions and outline future work.

2 Background

The most widely known clustering algorithm is the K-means clustering algorithm which was first published in 1955 [17]. Even today, k-means is still widely applied and researched in different fields because of its ease of implementation, simplicity, efficiency, and empirical success. K-means is a typical representative-based algorithm, which finds firstly the representative of each cluster, then assigns each object to its most similar representative, and at last forms the clusters with objects with the same representative [29]. On the other hand, k-means is a partitional clustering algorithm, which finds all the clusters simultaneously by partitioning all the objects, and does not have a hierarchical structure unlike hierarchical algorithms [17, 20].

It has long been shown that the performance of k-means depends greatly on the initialisation of means. Several initialisation methods were proposed for k-means, e.g. k-means++ [3]. Recent research shows that k-means probably reaches the global optimum when the initialisation means are well separated [17]. However, the usual way to overcome the local optima is still to run the k-means algorithm, given a k value, multiple times with different initial means and choose the clustering result with the smallest cost function [26].

As with many other machine learning algorithms, the basic K-means algorithm cannot directly deal with categorical data. Firstly, the common distance measure used in k-means is (squared) Euclidean distance, which can only be computed with numeric data. Secondly, the arithmetic mean is taken as the representative of each cluster, which is also a concept only available for numeric data. However, categorical data is as important as numeric data empirically, and this problem limits the usage of k-means considerably. In order to solve the problem and make k-means fit different data types, there are several ways to adapt k-means to deal with categorical data.

The traditional method for most machine learning algorithm to deal with categorical data is to convert all the categorical data into numeric data [26]. Ordinal data can be converted readily into numeric data easily based on its inherent order, but for truly nominal data it is impossible to order it in a meaningful way. The distance from “Red” to “Green” is the same as the distance to “Blue” or “Yellow”. Therefore, for nominal data, other methods are required. In this paper, two commonly used methods are considered 1-of-K coding and k-prototypes.

The first method is due to Ralambondrainy [22] who proposed an extended k-means algorithm as the complement for categorical data clustering. Before the normal k-means steps, this algorithm converts each multiple category feature into a set of binary features using 1 and 0 to represent a category value present or absent in objects. This method is also called 1-of-K coding and popularly adopted not only in k-means, but also in other machine learning algorithms, like kNN [12].

K-prototype on the other hand inherits the ideas of k-means, but applies different distances and different representatives to numeric data and categorical data [25]. For a dataset with both numeric and categorical features, the features can be organised as $A_1^n, A_2^n, ..., A_p^n, A_{p+1}^c, ..., A_m^c$, where m is the total amount of features, p is the amount of numeric features and $(m - p)$ is the amount of categorical features. K-prototype applies the same distance and representative to the first p numeric features, but for last $(m - p)$ categorical features, the limitations of k-means can be removed by the following modifications [15]:

1 using the simple matching distance for categorical features;
2 replacing means of clusters by modes.

Except for the definitions of distance and representative, k-prototypes inherits all the implementation process of k-means, so the simplicity and efficiency of k-means are well retained in k-prototypes. It is easy to be found that if the dataset only contains numeric features, k-prototypes is equal to k-means. For the situation that the dataset only contains categorical features, this algorithm can be considered as another algorithm called k-modes that can only deal with purely categorical data [26].

K-prototypes has become one of the most famous methods in categorical data clustering [25]. It is extended in many different ways and also used as the benchmark to be compared with. [2] discusses the initialisation methods of k-prototypes. In [16], k-modes is taken as one of the methods to generate base clusterings for categorical data. [4] presents an extension of the k-modes for clustering high-dimensional categorical data. [9] takes k-modes as benchmark and proposes a modified algorithm based on it. [10] presents an approximation algorithm to improve k-modes. [13] proposes the fuzzy k-modes algorithm.

Although both 1-of-K coding and k-prototypes have been widely used, the comparison of their performances has not been implemented systematically. From the theoretical perspective, there are some points of view, e.g. k-modes is faster because it needs less iterations to converge [15], 1-of-K coding requires more space and time for implementation because it largely expands the dimensionality [14], there is information loss in both methods [1, 15, 12, 25], and neither methods guarantee the global optimum [9, 10]. However, these points of views and to what degree they affect the clustering performance have not been examined by experiments. There are many reasons for this problem - it is too difficult to generate artificial datasets with categorical data for clustering [15], while there is not a mutual internal evaluation method to compare clustering algorithms defined with different distances. In this paper, we implement the empirical comparison with external evaluation but with new features designed especially for this purpose.

3 Comparison of 1-of-K Coding and K-prototypes

Normally, there are two types of measures that can be used to evaluate machine learning algorithms in empirical studies: external measures and internal measures [15, 29]. The former is based on labelled datasets as the ground truth, and compare the learning results with the existing labels to uncover how good the learning is. The latter focuses on the intrinsic structure and characteristics of datasets, rather than external man-made labels, so it is widely used in the evaluation of clustering problems, like choosing the best k value in k-means. Even though they can be calculated based on any distance, the internal measures, like the silhouette coefficient [23], cannot be used in the comparison between algorithms with different distances. Therefore, we use only the external measures in the present experiment to evaluate the clustering results.

Due to the limitation of resources, the ideal datasets from industry and with mature labels are not available. Besides, as discussed above, it is too difficult to generate artificial datasets with categorical data for clustering, our experiment is designed based on real world datasets with labels. All the datasets used here are from the UC Irvine Machine Learning Repository (`http://archive.ics.uci.edu/ml/`). We firstly choose 4 datasets that are famous and widely-used in the research of categorical data clustering: Soybean [15, 5, 16, 4, 19, 27], Congressional Voting Records [5, 16, 7, 9, 10], Credit Approval [15, 25, 11, 19] and Mushroom [5, 16, 4, 7, 9, 28, 10]. We note that the labels are made by human experts for a specific purpose, e.g. in the Credit Approval dataset, the data is the general information of people, but the labels are only about if the people were granted credit, which can only represent the data from a specific aspect, rather than the main structure of data. Therefore, we need to evaluate the dataset labels prior to the comparison so that we choose only the datasets with labels correlated with both results by k-means and k-prototypes. In addition, two large datasets, Adult and Bank Marketing, are added into the experiment for the evaluation of the time consumed during the clustering. Detailed information about the datasets is listed as Table. 1.

Table 1: The Real World Databases Selected for the Experiment

No.	**Datasets**	**Instances**	**Total Attributes**	**Type**
1	Soybean	47	35 (All Categorical)	Categorical
2	Congressional Voting Records	435	16 (All Categorical)	Categorical
3	Credit Approval	690	15 (9 Categorical + 6 Numeric)	Mixed
4	Mushroom	8124	22 (All Categorical)	Categorical
5	Adult	48842	14 (8 Categorical + 6 Numeric)	Mixed
6	Bank Marketing	45211	16 (9 Categorical + 7 Numeric)	Mixed

Because of the lack of expert knowledge about these data, all the instances with null value are removed before input. However, the instances with “?” instead of null value are retained, because it is considered as “unknown”, a category from the real-life situation. After the filtering, the datasets are re-organised into the format for clustering implementation.

Our experiments are implemented with each dataset and each algorithm as outlined in Fig. 1. For k-means, 1-of-K coding is implemented at the first stage, so that all the data can be normalised. For k-prototypes, there is no need to implement 1-of-K coding, but a range of the parameter γ are used for each dataset. For both algorithms, 100 runs will be implemented for each situation (different γ for k-prototypes). Although for purely categorical data, the setting of a range of the parameter γ does not affect the clustering result, we still use a range of γ for the evaluation of the clustering efficiency. During the process, there are four steps worth mentioning: the normalisation, the selection of k value, 100 runs for each situation, and the initialisation.

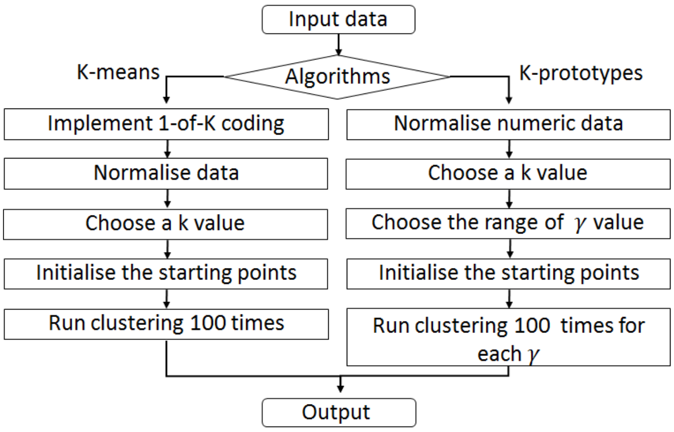


Fig. 1: Experimental Process

1 Normalisation

Lots of research has been conducted on normalisation methods. Based on Steinley's review paper [26], normalisation by range as Eq. 1, rather than z-scores, leads to better performances specially for k-means clustering.

$$X^{'} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

For k-prototypes, the definition of k-prototypes requires also normalisation by range for numeric data [15]. Therefore, normalisation by range is adopted in the experiment.

2 Selection of k

In the experiment, the k values are selected just as the labels show, that is, k equals to the number of different categories of the labels. The methods of selecting k in k-means have been discussed a lot in previous research [26], most of which can also be applied to k-prototypes.

3 100 runs for each situation

Due to the characteristics of k-means and k-prototypes, the global optimum is not guaranteed in a single run of clustering. The common way is to run it multiple times with the same parameter setting, and then choose the result with the best cost function as the final clustering result. Therefore, we only focus on a range of good results, rather than all of them, which is different from the evaluation of other machine learning applications. Likewise, the stability discussed in this paper is the concept how often the good results can be achieved, rather than the analysis of mean or variance of all the results in other applications.

4 Initialisation

For both k-means and k-prototypes, different local optima depend on the starting centroids (means or prototypes). Although lots of initialisation methods were proposed to avoid locally optimal solution [26], k-means++ [3] is the most popular method. In k-means++, only the first centroid is uniformly chosen from the data points in the dataset, and each subsequent centroid is chosen from the remaining data points with probability proportional to its squared distance to its closest existing centroid.

The evaluation of results starts from the conparison of efficiency by analysing the time, the number of iterations and the dimensionality of input. After that, external measures are used for the evaluation of clustering validity. There are plenty of external measure that are widely used in clustering evaluation, such as F-measure, Normalised Mutual Information, Jaccard Coefficient [29]. Accuracy [15, 13, 9, 27] is adopted in this experiment, because it is easy to understand and the k value is just from 2 to 4. The clustering accuracy r is defined as:

$$r = \frac{\sum_{i=1}^{k} a_i}{n} \tag{2}$$

where k is the number of clusters, n is the number of instances, and a_i is number of instances that are clustered correctly in this cluster. For different combinations of clustering results and the existing labels, clustering accuracy r is defined as the maximum value.

It should be noted that the accuracies of different datasets are not necessarily positively correlated with the validity of the clustering, because subjective opinions have been added into the labels when human experts labelled the datasets. On the other hand, the only measure we are sure to know how good the clustering is for each algorithm is the cost function. Therefore, before the evaluation of validity of algorithms, each dataset need to be checked if its accuracy results have the same trend as the cost function. Only the datasets that have accuracy results with the same trends as their cost functions of both algorithms can be used in the final evaluation of clustering validity. [3]

[3] The reason why we cannot just use cost function to compare the results is that the cost function is defined with different types of distances in different algorithms so the comparison with cost function will be meaningless.

4 Results

In this section we describe the results of our empirical analysis. We begin with a discussion of run time costs before moving on to consider measures of accuracy.

From the time consumed in 100 runs for each algorithm (Fig. 2 and Fig. 3), it is shown that when the dataset gets large, the time consumed for k-means is 2 to 3 times greater than that for k-prototypes. The time consumed in calculation may not reflect the genuine efficiency of algorithms exactly, but from the commercial perspective, it is meaningful that the implementation of k-prototypes is generally much faster than k-means.

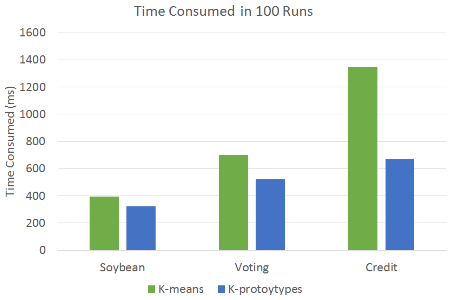


Fig. 2: Time Consumed - Soybean, Voting and Credit

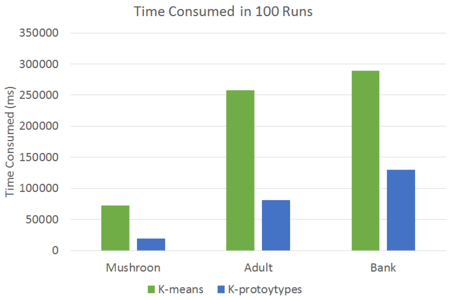


Fig. 3: Time Consumed - Mushroom, Adult and Bank

However, from the number of iterations in each run (Fig. 4) and the number of features before/after 1-of-K coding (Fig. 5), we can see that the k-means algorithm consumes much more time not because it needs more iterations to converge, but because 1-of-K coding substantially expands the dimensionality of the datasets.

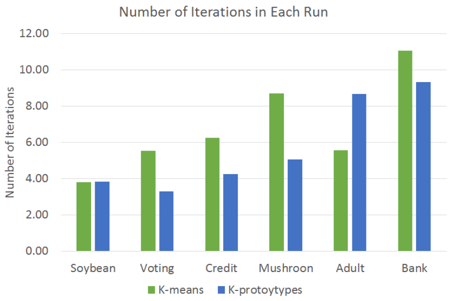


Fig. 4: Number of Iterations in Each Run

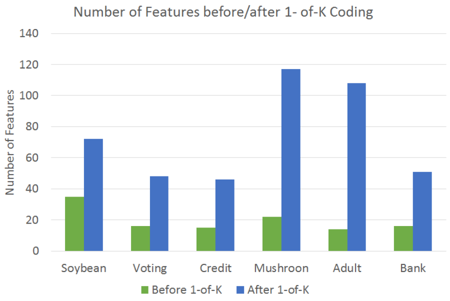


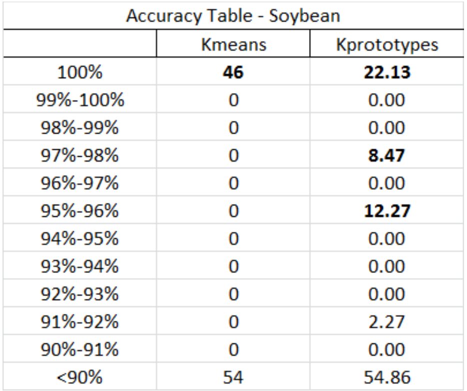
Fig. 5: Number of Features before/after 1-of-K Coding

As explained before, only the datasets that have accuracy results with the same trends as the cost functions of both algorithms can be used in the final evaluation of clustering validity. Among these 6 datasets as Table. 2, only 3 datasets are chosen: Soybean, Congressional Voting Records and Mushroom.

Table 2: The Correlation between Accuracy Results and Cost Functions

No.	**Datasets**	**Accuracy Correlation with**
1	**Soybean**	**Both**
2	**Congressional Voting Records**	**Both**
3	Credit Approval	K-prototypes
4	**Mushroom**	**Both**
5	Adult	K-means
6	Bank Marketing	None

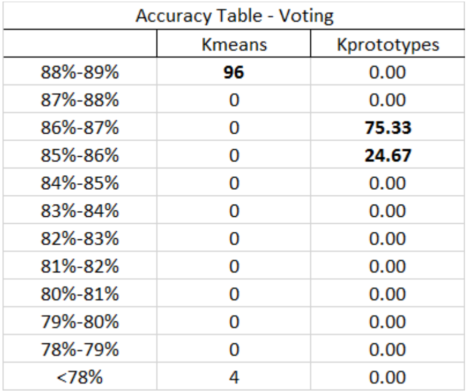
This however does not mean that the accuracy results of these 3 datasets have absolutely positive correlations with the cost function results. After all, they are not artificial datasets that are exactly designed for clustering. But the accuracy results of these 3 datasets have almost the same trends as cost function to show how good the clustering is, so they can be considered as the mediums between the two algorithms, so used in the evaluation.



Accuracy Table - Soybean

	Kmeans	Kprototypes
100%	**46**	**22.13**
99%-100%	0	0.00
98%-99%	0	0.00
97%-98%	0	**8.47**
96%-97%	0	0.00
95%-96%	0	**12.27**
94%-95%	0	0.00
93%-94%	0	0.00
92%-93%	0	0.00
91%-92%	0	2.27
90%-91%	0	0.00
<90%	54	54.86

Fig. 6: Accuracy Table - Soybean



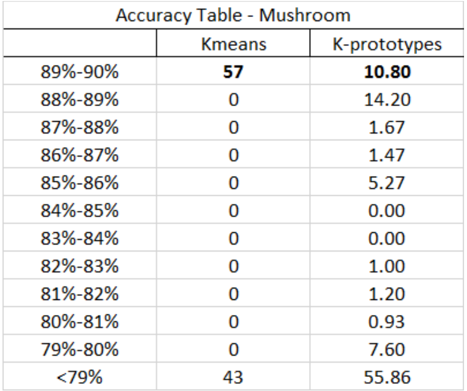
Accuracy Table - Voting

	Kmeans	Kprototypes
88%-89%	**96**	0.00
87%-88%	0	0.00
86%-87%	0	**75.33**
85%-86%	0	**24.67**
84%-85%	0	0.00
83%-84%	0	0.00
82%-83%	0	0.00
81%-82%	0	0.00
80%-81%	0	0.00
79%-80%	0	0.00
78%-79%	0	0.00
<78%	4	0.00

Fig. 7: Accuracy Table - Voting

Fig. 6, Fig. 7 and Fig. 8 summarise the accuracy calculation results of Soybean, Congressional Voting Records and Mushroom respectively. The first columns give the clustering accuracy intervals. The second and third columns show the numbers of clustering results that fall into a specific interval. There are in total 100 in each column. For k-prototypes, the experiment is implemented

100 runs with each γ, and the averages with decimals are filled into the table, because all of the datasets are purely categorical. From these tables, we get to compare the validity of these two algorithms.



Accuracy Table - Mushroom

	Kmeans	K-prototypes
89%-90%	**57**	**10.80**
88%-89%	0	14.20
87%-88%	0	1.67
86%-87%	0	1.47
85%-86%	0	5.27
84%-85%	0	0.00
83%-84%	0	0.00
82%-83%	0	1.00
81%-82%	0	1.20
80%-81%	0	0.93
79%-80%	0	7.60
<79%	43	55.86

Fig. 8: Accuracy Table - Mushroom

From these results we can make the following observations:

1 Both algorithms get almost the same highest accuracy. For Soybean and Mushroom, the differences are within 1%, while for Congressional Voting Records, the difference is about 2%;

2 If taking the best accuracy as BR, and the clustering whose results fall into the interval of $[BR-10\%, BR]$ as valid clustering, it is obvious that the valid results with k-means concentrate at the interval of highest accuracy, while the ones with k-prototypes spread much more widely in the valid interval, but the total numbers of valid clustering are not quite different. From this perspective, k-means is more stable than k-prototypes;

3 The numbers in bold refer to the best results based on cost function, that is, the objectively best clustering results. It is shown that for k-means all the best results in a situation lead to the same result with the best accuracy, but for k-prototypes, they may lead to multiple best results with even different performances. In other words, k-means probably finds only one global optimum, but k-prototype can find multiple global optima. Because the calculation in k-prototypes is based on integers, it generates the same cost function easily even when the clustering results are different, while this is very rare in k-means.

5 Conclusion

In this paper we have presented k-means with 1-of-K coding and k-prototypes as two valid clustering algorithms for categorical data.

Even though they use different distances in calculating dissimilarity, k-means with 1-of-K coding and k-prototypes provide similar best results. For the clustering speed, k-prototypes is faster than k-means with 1-of-K coding, because the latter expands significantly the dimensionality of the original dataset. For the clustering validity, because of the characteristics of each algorithm, the valid results with k-prototypes spread in multiple optima, while the ones with k-means with 1-of-K coding concentrate in one point. Therefore, we conclude that k-means with 1-of-K coding is more stable than k-prototypes.

Due to the preliminary nature of our studies and also the space constraints here, many questions about k-prototypes are not discussed in this paper, e.g., the selection of k value, the setting of parameter and the feature weighting. Each of these requires more research.

As a valid clustering algorithm for categorical data, k-prototypes can be explored in different ways. On one side, many extensions of k-means or other clustering algorithms can be adjusted and applied into k-prototypes, e.g. using the silhouette coefficient in clustering result evaluation, using Hopkins statistics to find the tendency of datasets for clustering and so on. On the other side, the idea of k-prototypes, especially the distance used in it, can be used directly to modify other algorithms and make them applicable in categorical data, e.g. density-based clustering algorithms. We see this as useful valid future research.

Acknowledgement. The authors wish to acknowledge the support of Enterprise Ireland through the Innovation Partnership Programme SmartSeg 2.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data & Knowledge Engineering 63(2), 503–527 (2007)
2. Ahmad, I.: K-Mean and K-Prototype Algorithms Performance Analysis. International Journal of Computer and Information Technology 03(04), 823–828 (2014)
3. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027–1035. Society for Industrial and Applied Mathematics (2007)
4. Bai, L., Liang, J., Dang, C., Cao, F.: A novel attribute weighting algorithm for clustering high-dimensional categorical data. Pattern Recognition 44(12), 2843–2861 (2011)
5. Bhagat, P.M., Halgaonkar, P.S., Wadhai, V.M.: Review of Clustering Algorithm for Categorical Data (2), 341–345 (2013)
6. Bishop, C.M.: Pattern recognition. Machine Learning (2006)
7. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. In: Data Engineering, 1999. Proceedings., 15th International Conference on. pp. 512–521. IEEE (1999)
8. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer 27(2), 83–85 (2005)
9. He, Z., Deng, S., Xu, X.: Improving k-modes algorithm considering frequencies of attribute values in mode. In: Computational Intelligence and Security, pp. 157–162. Springer (2005)

10. He, Z., Deng, S., Xu, X.: Approximation algorithms for k-modes clustering. In: Computational Intelligence, pp. 296–302. Springer (2006)
11. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: A cluster ensemble approach. arXiv preprint cs/0509011 (2005)
12. Hsu, C.C., Huang, W.H., et al.: Integrated dimensionality reduction technique for mixed data involving categorical values. In: Human Capital without Borders: Knowledge and Learning for Quality of Life; Proceedings of the Management, Knowledge and Learning International Conference 2014. pp. 245–255. ToKnow-Press (2014)
13. Huang, Z., Ng, M.K.: A Fuzzy k -Modes Algorithm for Clustering Categorical Data. IEEE Transactions on Fuzzy Systems 7(4), 446–452 (1999)
14. Huang, Z.: Clustering large data sets with mixed numeric and categorical values. In: Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD). pp. 21–34. Singapore (1997)
15. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data mining and knowledge discovery 2(3), 283–304 (1998)
16. Iam-On, N., Boongeon, T., Garrett, S., Price, C.: A link-based cluster ensemble approach for categorical data clustering. Knowledge and Data Engineering, IEEE Transactions on 24(3), 413–425 (2012)
17. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern recognition letters 31(8), 651–666 (2010)
18. Kelleher, J.D., Mac Namee, B., D'Arcy, A.: Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies. MIT Press (2015)
19. Kim, D.W., Lee, K.H., Lee, D.: Fuzzy clustering of categorical data using fuzzy centroids. Pattern Recognition Letters 25(11), 1263–1271 (2004)
20. Kotsiantis, S., Pintelas, P.: Recent advances in clustering: A brief survey. WSEAS Transactions on Information Science and Applications 1(1), 73–81 (2004)
21. MacKay, D.J.: Information theory, inference and learning algorithms. Cambridge university press (2003)
22. Ralambondrainy, H.: A conceptual version of the k-means algorithm. Pattern Recognition Letters 16(11), 1147–1157 (1995)
23. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53 – 65 (1987), http://www.sciencedirect.com/science/article/pii/0377042787901257
24. Sayal, R., Kumar, V.V.: A novel similarity measure for clustering categorical data sets. International Journal of Computer Applications 17(1), 25–30 (2011)
25. Shih, M.Y., Jheng, J.W., Lai, L.F.: A two-step method for clustering mixed categroical and numeric data. Tamkang Journal of science and Engineering 13(1), 11–19 (2010)
26. Steinley, D.: K-means clustering: A half-century synthesis. British Journal of Mathematical and Statistical Psychology 59(1), 1–34 (2006)
27. Sun, Y., Zhu, Q., Chen, Z.: An iterative initial-points refinement algorithm for categorical data clustering. Pattern Recognition Letters 23(7), 875–884 (2002)
28. Xu, R., Wunsch, D., et al.: Survey of clustering algorithms. Neural Networks, IEEE Transactions on 16(3), 645–678 (2005)
29. Zaki, M.J., Meira Jr, W.: Data mining and analysis: fundamental concepts and algorithms. Cambridge University Press (2014)