A Comparison of Categorical Attribute Data Clustering Methods

Ville Hautamäki[1], Antti Pöllänen[1], Tomi Kinnunen[1], Kong Aik Lee[2], Haizhou Li[2], and Pasi Fränti[1]

[1] School of Computing, University of Eastern Finland, Finland
[2] Institute for Infocomm Research, A*STAR, Singapore
villeh@cs.uef.fi

Abstract. Clustering data in Euclidean space has a long tradition and there has been considerable attention on analyzing several different cost functions. Unfortunately these result rarely generalize to clustering of categorical attribute data. Instead, a simple heuristic k-modes is the most commonly used method despite its modest performance. In this study, we model clusters by their empirical distributions and use expected entropy as the objective function. A novel clustering algorithm is designed based on local search for this objective function and compared against six existing algorithms on well known data sets. The proposed method provides better clustering quality than the other iterative methods at the cost of higher time complexity.

1 Introduction

The goal of *clustering* [1] is to reveal hidden structures in a given data set by grouping similar data objects together while keeping dissimilar data objects in separated groups. Let X denote the set of data objects to be clustered. The classical clustering problem setting considers data objects in a D-dimensional vector space, $X \subset \mathbb{R}^D$. The most commonly used objective function for such data is *mean squared error* (MSE). A generic solution is the well-known k-means method [2], which consists of two steps that are iterated until convergence. In *assignment step* (or E-step), all vectors are assigned to new clusters and *re-estimation step* (or M-step), model parameters are updated based on the new assigments.

Different from vector space data, data in educational sciences, sociology, market studies, biology and bioinformatics often involves *categorical attributes*, also known as *nominal* data. For instance, a data object could be a single questionnaire form that consists of multiple-choice questions. Possible outcomes of the answers can be encoded as integers. In this way, each questionnaire would be represented as an element of $\mathbb{N}^D$, where D is the number of questions. Unfortunately, since, the categories do not have any natural ordering, applying clustering methods developed for metric space data cannot be applied as such.

Hamming distance is a distance function designed for categorical data. It counts the number of attributes where two vectors disagree, i.e., having different

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 53–62, 2014.

attribute values. Cost functions and algorithms based on Hamming distance include *k-medoids* [3] and *k-modes* [4], both being extensions of the classical k-means [2]. In k-medoids, cluster representative (*discrete median*) is a vector in the cluster that minimizes the sum of distances from all other vectors to the cluster representative. In k-modes, the representative is the *mode* of the cluster, calculated independently for every attribute. Mode is the most frequently occurring value, in one attribute, over all the vectors in the cluster.

Using minimum Hamming distance as the assignment rule, one is also faced with the so-called zero probability condition [5]. It is one of the the assumptions behind the convergence proof of the classical k-means algorithm, stating that that the probability of assigning a vector to more than one cluster must be zero. With real valued data this condition holds. However, in tne case of categorical attribute clustering based on Hamming distance this condition is clearly not met. In the extreme case, when two D-dimensional vectors are maximally different, their Hamming distance is D. Consequently, the Hamming distance can take up only D unique values and it is likely that a vector is equally close to more than one cluster. Moreover, in the k-modes method, the cluster representative (mode) is not unique either. Tie-breaking needs to be employed in both the E- and the M-steps.

Tie-breaking problem in the cluster assignment (E-step) phase can be solved by testing each vector one by one whether its move to a new cluster will improve the objective function value. If such a cluster is found, the cluster parameters are immediately updated. Convergence of the algorithm can then be detected when there is no movement of vectors. One way to tackle the tie-breaking problem in the M-step is to represent the cluster by its *probability mass function* (pmf), that is, the relative frequencies of each category value in the cluster. For example, choices for educational background could have values $P(\text{elementary school}) = 0.2$, $P(\text{high school}) = 0.7$ and $P(\text{vocational school}) = 0.1$. In a sense, k-modes can be considered as a quantized version of the pmf-based cost functions. In this example, "high school" would be the cluster representative.

A number of different objective functions have been proposed, based on the the idea of modeling each cluster by its pmf: *k-histograms* [6, 7], *k-distributions* [8], *minimum description length* (MDL) [9], *mutual information* [10] and *expected entropy* [11–14]. Expected entropy is the average entropy of the entire clustering. If the pmf of the cluster is sharply peaked, its entropy is small. Therefore minimizing the expected entropy leads to compact clusters.

Despite the availability of multiple pmf-based methods, it is unclear which objective function and method would be best suited for a given application. In this work, we compare six well known categorical clustering methods in diverse categorical data sets using expected entropy as a clustering quality measure. Data sets vary from small sets of only 47 data points to large data set of more than 47k entries. In addition, we propose a new local search algorithm that directly optimizes the expected entropy.

2 Modeling Cluster by Its Distribution

In hard clustering, the goal is to divide a data set X, of size N, into disjoint clusters $\mathcal{V} = \{V_1, V_2, \ldots, V_M\}$, where $V_i \subset X, \cup_{i=1}^{M} V_i = X$, and $V_i \cap V_j = \emptyset \quad \forall\, i \neq j$.

In categorical clustering, data set consists of vectors $\boldsymbol{x} = (x_1, x_2, \ldots, x_D)$, where each x_d takes values from a discrete set (categories). The number of categories in dimension d is denoted by C_d. We assume, without a loss of generality that $x_d \in \{1, \ldots, C\}$, where $C = \max_{d=1\ldots D} C_d$.

Entropy [15] is a measure of "surprise" in the data. High entropy signifies flat distribution whereas low entropy signifies peaked distribution. Formally, entropy for discrete distribution is defined as:

$$H(X) \triangleq -\sum_{\boldsymbol{x} \in X} p(\boldsymbol{x}) \log p(\boldsymbol{x}), \tag{1}$$

where $p(\boldsymbol{x}) = p(x_1, \ldots, x_D)$. Here, $p(\boldsymbol{x})$ denotes *estimated probability* of the joint event $(x_1, \ldots, x_D)$. In the rest of the discussion, by entropy we will mean *estimated entropy*, also known as *empirical entropy*.

Our goal is to minimize the so-called *expected entropy*) [12]:

$$H(\mathcal{V}) \triangleq \sum_{m=1}^{M} \frac{|V_m|}{N} H(V_m), \tag{2}$$

where $|V_m|$ is the cardinality of V_m and $H(V_m)$ is the entropy of the cluster V_m. Note that by setting $M = 1$, we obtain $H(\mathcal{V}) = H(X)$, and by setting $M = N$, we obtain $H(\mathcal{V}) = 0$, where each vector is in its own cluster. All other values are between these two extremes.

3 Algorithms

We evaluate two different types of clustering approaches, *iterative* and *agglomerative*. In iterative algorithms, clustering cost is improved in each iteration by repartitioning the datasets. The selected algorithms are summarized in Table 1. In agglomerative algorithms, instead, clusters are merged one by one until a desired number of clusters is reached. Two agglomerative methdos are considered: ACE [14], which optimizes the expected entropy (2), and ROCK [16] which optimizes its own cost function.

3.1 Prototype Based Algorithms

Prototype-based iterative methods [3, 4] select one vector from each cluster as a representative, analogous to centroid vector in conventional k-means. The k-modes and k-medoid methods use minimum Hamming distance to assign vectors to clusters. In the classical k-means, squared Euclidean distance was used. In the M-step, the goal is to find such a prototype per cluster that minimizes Hamming distance from each vector in the cluster to the prototype vector. In k-medoid, one vector from the cluster is selected as the prototype and in k-modes, most frequently observed category per dimension is selected.

Table 1. Summary of k-means type methods experimented in this study, classified according to cluster representative type and distance measure

Method	Representative	Measure
k-distributions [8]	Distribution	Product of m-estimates
k-histograms [6, 7]	Distribution	non-matching frequencies
k-modes [4]	Mode	Hamming distance
k-medoids [3]	Medoid	Hamming distance
k-entropies [**this paper**]	Distribution	Entropy change

3.2 k-Distributions

In k-distributions [8], there are no cluster prototypes but the histograms are used to represent clusters. In the E-step, a vector is assigned to the cluster that maximizes the likelihood $p(\boldsymbol{x}|V_m)$. The likelihood can be factorized into each dimension separately assuming that dimensions are independent. Some categories may have zero count, the histogram is therefore processed by *Laplacian smoothing* [17].

The expected entropy is not directly optimized by k-distributions. No proof of convergence exists, but experimentally we have noticed that the method seems to converge, albeit slowly. In the following, we attempt to give an explanation of the slow convergence. It would benefit if the similarity measure between a vector and cluster remains relatively stable when only small changes are made in the cluster partitioning. Unfortunately, this is not the case with k-distributions. Let us consider a case where we map a vector to a cluster, where one dimension has a non-matching category (no vector in the cluster has that category). When a new vector having this non-matching category is added to the cluster, comparing likelihood before and after addition we notice a large difference. For example, the likelihood from a vector after addition of the cluster with 15 vectors and 3 categories is 3.5 times more than before the addition. Thus, vectors end up changing clusters very often, leading to a slow convergence.

3.3 k-Representatives and k-Histograms

K-representatives [7] first assigns randomly all vectors to clusters and computes normalized histograms as representatives of each cluster. Frequencies are normalized so that they sum up to one. The distance measure from vector to cluster is Hamming distance weighted by the frequency. The method assigns new vectors to clusters based on the distance measure and recomputes the histograms. Process continues until no re-assignments of vectors are detected.

Unfortunately, contrary to the claim in [7], we found out that algorithm does not always converge[1], We, therefore, do not consider k-representatives method

[1] Proof by explicit construction of a 5-dimensional data set that k-representatives does not converge: `http://cs.uef.fi/sipu/krepresentatives.pdf`.

further. In iterative clustering with immediate update, vector is moved from one cluster to another if the move *decreases* the cost function. We considere here k-histograms [6] cost, which is the sum of k-representatives distance measures. It is a non-negative cost function, thus, the algorithm converges in a finite number of steps. The k-histograms method uses the immediate update strategy, otherwise it is the same as k-representatives.

3.4 Agglomerative Methods

A *robust clustering algorithm for categorical attributes* (ROCK) [16] defines a cost function based on the idea of neighbours and links. Neighbourhood of each vector is decided based on thresholded distance between vectors. We use Hamming distance. A link between two vectors is made if they share at least one neighbour. The goal of ROCK is to maximize pairwise links between vectors within the clusters, and minimize links between clusters. In each iteration, ROCK merges two the clusters that maximizes this criterion.

In *Agglomerative Categorical clustering with Entropy criterion* ACE method [14], expected entropy is optimized. In each iteration, ACE merges two clusters, V_i and V_j, so that the *incremental entropy* is minimized:

$$I_m(V_i, V_j) = H(V_i \cup V_j) - H(V_i) - H(V_j). \tag{3}$$

3.5 The Proposed Method

We propose to optimize the expected entropy directly. We start by randomly assigning each vector to a cluster. The method then iterates over all vectors and tests whether moving it to a new cluster improves the expected entropy. The assignment that maximally improves is selected. The algorithm converges when no vector changes its cluster assignment. It is easy to see that this strategy converges as each iteration is forced to either improve on the previous solution, or keep the existing one and stop. The proposed method is summarized in Algorithm 1.

Algorithm 1. The proposed method (k-entropies)

```
Randomly assign all vectors to M clusters.
Model clusters as their probability mass function (pmf).
repeat
  for x ∈ X do
    V_i ← Assign x according to minimum cost (2).
    Estimate prototype of the cluster V_i as the pmf of the cluster.
  end for
until No change in vector assignments.
```

3.6 Summary

All the algorithms, mentioned above are summarized in Table 2. The time and space complexities for ACE and ROCK are referenced from the respective publications, and the others have been derived by ourselves. The quadratic space and time complexity of both ACE and ROCK makes them rather impractical for large data sets. Here, I denotes the number iterations, C_{avg} the average number of categories, T_L the cost of computing logarithm, $R_{\max}$ the maximum number of neighbours, and R_{avg} the average number of neighbours.

Table 2. Summary of clustering algorithms

Algorithm	Type	Time complexity	Space complexity
ACE [14]	Agglomerative	$O(N^2 \log N + N^2 D C_{\text{avg}})$	$O(N^2)$
ROCK [16]	Agglomerative	$O(N^2 \log N + N^2)$	$O(\min\{N^2, N R_{\max} R_{\text{avg}}\})$
k-medoids [3]	k-means	$O(INMD)$	$O(N)$
k-modes [4]	k-means	$O(INMD)$	$O(N)$
k-distributions [8]	k-means	$O(INMDT_L)$	$O(N)$
k-histograms [6]	Immediate update	$O(INMD)$	$O(N)$
k-entropies	Immediate update	$O(INMDC_{\text{avg}}T_L)$	$O(N)$

4 Experiments

Experimental comparison were performed using six different categorical data sets (Mushroom, Votes, Soybean, CENSUS, SPECT hearth and Plants) obtained from UCI Machine Learning archive [18]. Data sets are summarized in Table 3.

Only two methods optimize directly the expected entropy: ACE and k-entropies (proposed method). We are interested to find out how the other methods perform in terms of expected entropy as a clustering objective function, where low entropy is desired. For iterative schemes, the number of iterations I depends on initialization, data set and cost function, it can have importance on how fast the algorithm is in practice. Number of iterations I measures the empirical convergence speed of the algorithm.

Mushroom data set includes 8124 observations from 23 different mushroom species. it has 21 attributes, and 119 categories. Dimensions with large number of missing values were discarded in our tasks. Dimensions of the vectors encode forms and colours of the mushrooms. **Congressional votes** data set includes votes from the US Congress of 1984. Each vector gives the votes of one of the 435 member of the US congress. In total, proposals were collected with possible outcome of {yes, no, other}, where other means that politicians opinion on the said proposal is not known. Total number of categories is 46. **Soybean** data set contains observations from different soybeans. It contains 47 vectors, 35 dimensions and 72 categories. **CENSUS** data set is selected to evaluate scalability of

the compared methods. Data set size is 2,458,285, has 68 dimensions and 396 categories. This data set contains both nominal and ordinal data types. In our experiments, special processing for ordinal data is not used. **SPECT hearth** is data on cardiac *single proton emission computed tomography* (SPECT) images. Each SPECT image was summarized to 22 binary pattern features. Data set contains 267 patients, and 44 categories. **Plants** data set is transaction data about different growth locations, containing 34781 vectors (plants), 70 dimensions and 140 categories.

Table 3. Data set summary

Data set	Vectors	D	Categories	Entropy
Mushroom	8124	21	119	21.44
Votes	435	16	46	13.98
Soybean	47	35	72	18.80
CENSUS	2458285	68	396	55.17
SPECT hearth	267	22	44	13.68
Plants	34781	70	140	25.35

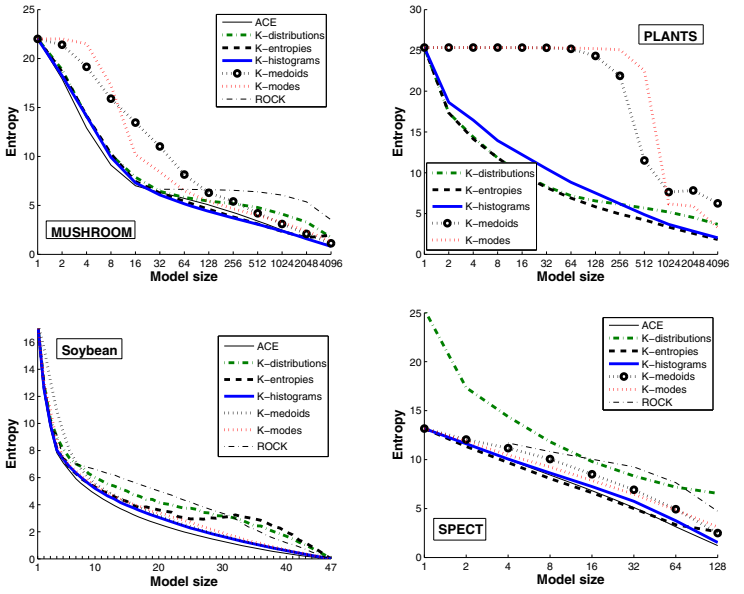

Fig. 1. Expected entropy as a function of model size. In order from top left to bottom right: mushroom, plants, soybean and spect data sets.

4.1 Quality of Clustering

Fig. 1 shows the expected entropy as a function of model size. First glance validates our intuition: methods that are based on optimizing distribution perform similarly. In general, the order of performance is: ACE first, then k-entropies and after that k-histograms. K-distributions gives different results for SPECT data set, when comparing to other sets.

The prototype based methods, k-medoids and k-modes optimize sum of Hamming distances and perform similarly, as expected. They cluster to the mushroom and plants data sets differenty than the pmf-based methods. ROCK also seems to follow its own trend. If no links exist between two clusters, then there is no way to merge them. This behaviour is visible in mushroom and SPECT data sets, in smaller model sizes ROCK is not able to obtain any results. Plants is transaction data, where most attributes have zero values, resulting all zero vector as a prototype with k-modes and k-medoids.

4.2 Summary of Experiments

Summary of average expected entropies and processing times with standard deviations is shown in Table 4 and 5, when repeating all experiments 10 times.

Table 4. Summary obtained average expected entropies and standard deviations.

Algorithm	Soybean $M=4$		Mushroom $M=16$		Votes $M=2$		SPECT $M=8$		Plants $M=1024$		CENSUS $M=16$	
	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std	$H(\mathcal{V})$	std
ACE	**7.83**	**0**	**7.01**	n/a	9.79	0.16	8.43	0.07	n/a	n/a	n/a	n/a
ROCK	9.20	**0**	n/a	n/a	**9.30**	**0**	10.82	**0**	n/n	n/a	n/a	n/a
k-medoids	10.94	1.57	13.46	0.71	10.00	0.95	10.06	0.44	7.64	0.33	32.61	0.87
k-modes	8.66	1.06	10.19	1.45	9.70	0.03	9.25	0.29	6.19	0.18	31.00	1.26
k-distributions	9.00	0.93	7.87	0.44	9.59	0.01	8.25	0.13	5.19	0.09	28.58	**0.27**
k-representatives	8.30	0.83	7.51	0.30	9.60	0.01	8.64	0.13	n/a	n/a	n/a	n/a
k-histograms	8.04	0.53	7.31	**0.18**	9.60	0.01	8.64	0.15	3.67	**0.02**	29.17	0.48
k-entropies	8.15	0.56	7.31	**0.18**	9.58	**0**	**8.06**	0.07	**3.33**	0.05	**28.51**	0.50

Table 5. Summary obtained average processing times (in seconds) and standard deviations

Algorithm	Mushroom		Plants		CENSUS	
	Time	std	Time	std	Time	std
ACE	2565.76	n/a	n/a	n/a	n/a	n/a
ROCK	n/a	n/a	n/a	n/a	n/a	n/a
k-medoids	0.06	0	30.89	8.12	184.95	18.63
k-modes	0.08	0.01	63.60	10.75	188.65	24.17
k-distributions	1.83	0.17	5043.80	1637.70	1748.34	530.97
k-representatives	0.30	0.07	n/a	n/a	n/a	n/a
k-histograms	0.19	0.04	467.46	110.44	900.35	81.81
k-entropies	12.51	1.43	8669.55	1000.84	6068.72	1116.87

Entry with n/a means that algorithm was not able to produce a result for that configuration, either due to non-convergence or running out of memory. Model sizes were selected for each data set separately, either by looking at the expected entropy as function of model size plot, or by information from the data set descriptions. For the plants data set we selected 1024, because for smaller model sizes k-modes and k-medoids completely fail.

We notice that for Soybean and Mushroom data sets ACE is the best as it directly optimizes the expected entropy. However, for the votes and SPECT data sets the proposed method provides better clustering, than ACE. The usability of ACE and ROCK are limited to their space complexity: those methods are not able to cluster largest sets at all. The proposed method is the best in terms of quality for the SPECT, plants and CENSUS data sets.

K-representatives results were also obtained for illustrative purposes for the datasets it converged on. It is slower than k-histograms, which can be attributed to the non-convergence behaviour of the algorithm. In terms of expected entropy, k-representatives iteration strategy did not provide any visible advantage over the immediate update of the k-histograms.

When comparing the proposed method and ACE in terms of processing time, we see that the proposed method is a clear winner. However, other methods that do not directly optimize expected entropy are clearly much faster.

5 Conclusions

We have compared existing pmf-based categorical clustering methods and found them to be very similar in terms of expected entropy. We also found out that the prototype-based methods (k-medoids and k-modes), while being the fastest methods, are not able to reach the lowest expected entropy obtained by the pmf-based methods. Thus, those methods are not recommended for clustering categorical data sets. On the other hand, ACE, while providing the best overall results, is not well-suited for large data sets, because of its quadratic time and space complexities. The proposed k-entropies method yielded the best results for the larger datasets. As a future work, we plan to investagate ways to obtain a k-means type clustering algorithm for the expected entropy cost.

Acknowledgements. This work was supported by Academy of Finland (projects 253000 and 253120).

References

1. Xu, R., Wunsch, D.: Survey of clustering algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
2. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. I, pp. 281–297. University of California (1967)

3. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: an introduction to cluster analysis. John Wiley Sons, New York (1990)
4. Huang, Z.: Extensions to k-means algorithm for clustering large data sets with categorical values. Data Mining Knowledge Discovery 2(3), 283–304 (1998)
5. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publisher, Boston (1992)
6. He, Z., Xu, X., Deng, S., Dong, B.: K-histograms: An efficient clustering algorithm for categorical dataset. CoRR abs/cs/0509033 (2005)
7. San, O.M., Huynh, V.N., Nakamori, Y.: An alternative extension of the k-means algorithm for clustering categorical data. International Journal of Applied Mathematics and Computer Science 14(2), 241–247 (2004)
8. Cai, Z., Wang, D., Jiang, L.: K-distributions: A new algorithm for clustering categorical data. In: Huang, D.-S., Heutte, L., Loog, M. (eds.) ICIC 2007. LNCS (LNAI), vol. 4682, pp. 436–443. Springer, Heidelberg (2007)
9. Chakrabarti, D., Papadimitrou, S., Modha, D.S., Faloutsos, C.: Fully automatic cross-associations. In: Proceedings of the ACM SIGKDD Conference (2004)
10. Andritsos, P., Tsaparas, P., Miller, R.J., Sevcik, K.C.: LIMBO: Scalable clustering of categorical data. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 123–146. Springer, Heidelberg (2004)
11. Barbará, D., Li, Y., Couto, J.: Coolcat: an entropy-based algorithm for categorical clustering. In: CIKM 2002: Proceedings of the Eleventh International Conference on Information and Knowledge Management, pp. 582–589. ACM, New York (2002)
12. Li, T., Ma, S., Ogihara, M.: Entropy-based criterion in categorical clustering. In: ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning, p. 68. ACM, New York (2004)
13. Li, T.: A unified view on clustering binary data. Machine Learning 62(3), 199–215 (2006)
14. Chen, K., Liu, L.: The "best k" for entropy-based categorical data clustering. In: Proceedings of the 17th International Conference on Scientific and Statistical Database Management (SSDBM 2005), Berkeley, USA, pp. 253–262 (2005)
15. Cover, T., Thomas, J.: Elements of Information Theory. Wiley-Interscience (1991)
16. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. Information Systems 25(5), 345–366 (2000)
17. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
18. Asuncion, A., Newman, D.: UCI machine learning repository (2007)