Clustering of Mixed-type Data Considering Concept Hierarchies

Sahar Behzadi[1], Nikola S. Müller[2], Claudia Plant[1,3], and Christian Böhm[4]

[1] Faculty of Computer Science, Data Mining, University of Vienna, Vienna, Austria
[2] Institute of Computational Biology, Helmholtz Zentrum München, Germany
[3] ds:UniVie, University of Vienna, Vienna, Austria
[4] Ludwig-Maximilians-Universität München, Germany

Abstract. Most clustering algorithms have been designed only for pure numerical or pure categorical data sets while nowadays many applications generate mixed data. It arises the question how to integrate various types of attributes so that one could efficiently group objects without loss of information. It is already well understood that a simple conversion of categorical attributes into a numerical domain is not sufficient since relationships between values such as a certain order are artificially introduced. Leveraging the natural conceptual hierarchy among categorical information, concept trees summarize the categorical attributes. In this paper we propose the algorithm *ClicoT* (**CL**ustering mixed-type data **I**ncluding **CO**ncept **T**rees) which is based on the Minimum Description Length (MDL) principle. Profiting of the conceptual hierarchies, ClicoT integrates categorical and numerical attributes by means of a MDL based objective function. The result of ClicoT is well interpretable since concept trees provide insights of categorical data. Extensive experiments on synthetic and real data set illustrate that ClicoT is noise-robust and yields well interpretable results in a short runtime.

1 Introduction

Clustering mixed-data is a non-trivial task and typically is not achieved by well-known clustering algorithms designed for a specific type. It is already well-understood that converting one type to another one is not sufficient since it might lead to information loss. Moreover, relations among values (e.g. a certain order) are artificially introduced. Let Figure 1 show a mixed-type data where three different clusters are illustrated by different shapes. The data set comprises of two numerical attributes concerning the position of objects and a categorical attribute representing the color. We simply converted the color to a numerical attribute by mapping numbers to various colors. Considering the *Normalized Mutual Information* (NMI) [12] as an evaluation measure, Figure 1 depicts the inefficiency of applying K-means and DBSCAN, two popular clustering algorithms, on the converted data. Therefore, integrating categorical and numerical attributes without any conversion is required since it preserves the original format of any attribute.

Utilizing the MDL principle we regard the clustering task as a data compression problem so that the best clustering is linked to the strongest data set

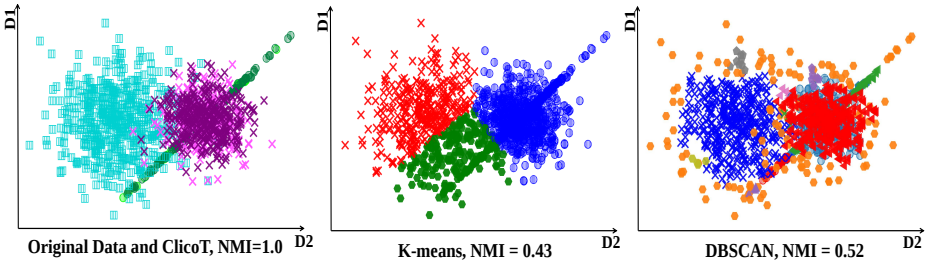

Fig. 1. Clustering results after converting categorical attribute *Color* to numerical.

compression. MDL allows integrative clustering by relating the concepts of likelihood and data compression while for any attribute a representative model is required. Although for solely numerical data sets a *Probability Distribution Function* (PDF) represents an approximation of data, finding an appropriate approximation for categorical attributes is not straight-forward. Considering the natural hierarchy among categorical values we introduce *concept hierarchy* to summarize the categorical information. Back to the running example, considering pink as a higher-level hierarchy for the objects in the cluster consisting of rose and purple points with the shape $\times$, more accurately represents the characteristics of the cluster.

Beyond the clustering approaches, detecting the most relevant attributes during this process improves the quality of clustering. However, considering a data set with an unknown distribution where only few subgroups in the data space are actually relevant to characterize a cluster, it is not trivial to recognize the cluster-specific attributes. Thus, we employ an information-theoretic greedy approach to specify the most relevant attributes. As a result, our novel parameter-free **CL**ustering algorithm for mixed-type data **I**ncluding **CO**ncept **T**ress, shortly *ClicoT*, provides a natural interpretation avoiding any conversion which leads to an effective clustering (c.f. Figure 1). Our approach consists of several contributions:

- **Integration:** ClicoT integrates two types of information considering data compression as an optimization goal. ClicoT flexibly learns the relative importance of the two different sources of information for clustering without requiring the user to specify input parameters which are usually difficult to estimate.
- **Interpretation:** In contrast to most clustering algorithms, ClicoT not only provides information about *which* objects are assigned to which clusters, but also gives an answer to the central question *why* objects are clustered together. As a result of ClicoT, each cluster is characterized by a signature of cluster-specific relevant attributes providing appropriate interpretations.
- **Robustness:** The compression-based objective function ensures that only the truly relevant attributes are marked as cluster-specific attributes. Thereby, we avoid over-fitting, enhance the interpretability and guarantee the validity of the result.
- **Usability:** ClicoT is convenient to be used in practice since our algorithm scales well to large data sets. Moreover, our compression-based approach avoids difficult estimation of input parameters e.g. the number or the size of clusters.

2 Clustering Mixed Data Types

To design a mixed-type clustering algorithm we need to address three fundamental questions: How to model numerical attributes to properly characterize a cluster? How to model categorical attributes? And finally how to efficiently integrate heterogeneous attributes when the most relevant attributes are specified? In principle, a PDF summarizes values by approximating meaningful parameters. However, the idea of using a background PDF for categorical attributes is not intuitive at first, therefore we employ concept hierarchies.

2.1 Concept Hierarchy

As mentioned, concept hierarchies allow us to express conceptual interchangeable values by selecting an inner node of a concept hierarchy to describe a cluster. Concept hierarchies not only capture more relevant categories for each cluster but also help to interpret the clustering result appropriately. Let $\mathcal{DB}$ denote a database consisting of n objects. An object o comprises m categorical attributes $\mathcal{A} = \{A_1, A_2, ..., A_m\}$ and d numerical attributes $\mathcal{X} = \{x_1, x_2, ..., x_d\}$. For a categorical attribute A_i, we denote different categorical values by ${A_i}^{(j)}$. An *Element* represents a categorical value or a numerical attribute and we denote the number of all *Element*s by E. Considering the natural hierarchy between different categories, for each categorical attribute A_i a concept hierarchy is already available as follows:

Definition 1. ***Concept Hierarchy.*** *Let $T_{A_i} = (N, \mathcal{E})$ be a tree with root A_i denoting the concept hierarchy corresponding to the categorical attribute A_i with the following properties:*

1. *T_{A_i} consists of a set of nodes $N = \{n_1, ..., n_s\}$ where any node is corresponding to a categorical concept. $\mathcal{E}$ is a set of directed edges $\mathcal{E} = \{e_1, ..., e_{s-1}\}$, where n_j is a parent of n_z if there is an edge $e_l \in \mathcal{E}$ so that $e_l = (n_j, n_z)$.*
2. *The level $l(n_j)$ of a node n_j is the height of the descendant sub-tree. If n_j is a leaf, then $l(n_j) = 0$. In a concept tree leaf nodes are categorical values existing in the dataset. The root node is the attribute A_i which has the highest level, also called the height of the concept hierarchy.*
3. *Each node $n_j \in N$ is associated with a probability $p(n_j)$ which is the frequency of the corresponding category in a dataset.*
4. *Each node n_j represents a sub-category of its parent therefore all probabilities of the children sum up to the probability of the parent node.*

2.2 Cluster-specific elements

Beside an efficient clustering approach, finding relevant attributes to capture the best fitting model is important. Usually the clustering result is disturbed by irrelevant attributes. To make the model for each cluster more precise we distinguish between relevant and irrelevant attributes. Each cluster c is associated with a subset of the numerical and categorical relevant elements denoted

by *cluster-specific elements*. Categorical cluster-specific elements are represented by a specific concept hierarchy which diverges from the background hierarchy (i.e. the concept hierarchy of the entire database).

Definition 2. ***Cluster.*** *A cluster c is described by:*

1. *A set of objects $\mathcal{O}_c \subset \mathcal{DB}$.*
2. *A cluster-specific subspace $I = \mathcal{X}_c \cup \mathcal{A}_c$, where $\mathcal{X}_c \subseteq \mathcal{X}$ and $\mathcal{A}_c \subseteq \mathcal{A}$.*
3. *For any categorical attribute $A_i \in \mathcal{A}_c$, the corresponding cluster-specific concept hierarchy is a tree ${T^c}_{A_i} = (N_c, \mathcal{E}_c)$ with nodes and edges as specified in Definition 1. $N_c \subset N$ indicates the cluster-specific nodes. For computing the probabilities associated with the cluster-specific nodes instead of all n objects, only the objects $\mathcal{O}_c$ in cluster c are applied, i.e. $p(n_j) = \frac{|n_j|}{|\mathcal{O}_c|}$.*

2.3 Integrative Objective Function

Given the appropriate model corresponding to any attribute, MDL allows a unified view on mixed data. The better the model matches major characteristics of the data, the better the result is. Following the MDL principle [11], we encode not only the data but also the model itself and minimize the overall description length. Simultaneously we avoid over-fitting since the MDL principle tends to a natural trade-off between model complexity and goodness-of-fit.

Definition 3. ***Objective Function.*** *Considering the cluster c the description length (DL) corresponding to this cluster defined as:*

$$DL(c) = DL_c(\mathcal{X}) + DL_c(\mathcal{A}) + DL(model(c))$$

The first two terms represent coding costs concerning numerical and categorical attributes, respectively while the last term is the model encoding cost. Our proposed objective function minimizes the overall description length of the database which is defined as:

$$DL(\mathcal{DB}) = \sum_{c \in \mathcal{C}} DL(c)$$

Coding Numerical Attributes: Considering Huffman coding scheme, the description length of a numerical value o_i is defined by $-\log_2 \mathrm{PDF}(o_i)$. We assume the same PDF to encode the objects in various clusters and clusters compete for an object while the description length is computed by means of the same PDF for evrey cluster. Therefore any PDF would be applicable and using a specific model is not a restriction [3]. For simplicity we select Gaussian PDF, $\mathcal{N}(\mu, \sigma)$. Moreover, we distinguish between the cluster-specific attributes in any cluster c, denoted by $\mathcal{X}_c$, and the remaining attributes $\mathcal{X} \setminus \mathcal{X}_c$ (Definition 2). Let μ_i and σ_i denote the mean and variance corresponding to the numerical attribute x_i in cluster c. If x_i is a cluster-specific element ($x_i \in \mathcal{X}_c$), we consider

only cluster points to compute the parameters otherwise ($x_j \in \mathcal{X} \setminus \mathcal{X}_c$) the overall data points will be considered. Thus, the coding cost for numerical attributes in cluster c is provided by:

$$DL_c(\mathcal{X}) = \sum_{x_i \in \mathcal{X}} \sum_{o_i \in \mathcal{O}_c} -\log_2 \Big(\mathcal{N}(\mu_i, \sigma_i)\Big)$$

Coding Categorical Attributes: Analogously, we employ Huffman coding scheme for categorical attributes. The associated probability to a category is its frequency w.r.t. either the specific or the background hierarchy (Definition 1). Similar to numerical attributes, we assume $\mathcal{A}_c$ as the set of cluster-specific categorical attributes and $\mathcal{A} \setminus \mathcal{A}_c$ for the rest. Let o_j denote a categorical object value corresponding to the attribute A_j. We define $f(A_j, o_j)$ as a function which maps o_j to a node in either a specific or a background hierarchy depending on A_j. Thus, the categorical coding cost for a cluster c is given by:

$$DL_c(\mathcal{A}) = \sum_{A_j \in \mathcal{A}} \sum_{o_j \in \mathcal{O}_c} -\log_2 \left(p(f(A_j, o_j))\right)$$

Model Complexity: Without taking the model complexity into account, the best result will be a clustering consisting of singleton clusters. This result is completely useless in terms of the interpretation. Focusing on cluster c, the model complexity is defined as:

$$DL(model(c)) = idCosts(c) + SpecificIdCosts(c) + paramCosts(c)$$

The idCosts are required to specify which cluster is assigned to a object while balancing the size of clusters. Employing the Huffman coding scheme, idCosts are defined by $|\mathcal{O}_c| \cdot log_2 \frac{n}{|\mathcal{O}_c|}$ where $|\mathcal{O}_c|$ denotes the number of objects assigned to cluster c. Moreover, in order to avoid information loss we need to specify whether an attribute is a cluster-specific attribute or not. That is, given the number of specific elements s in cluster c, the coding costs corresponding to these elements, *SpecificIdCosts*, is defined as:

$$SpecificIdCosts(c) = s \cdot \log_2 \frac{E}{s} + (E - s) \cdot \log_2 \frac{E}{(E - s)}$$

Following fundamental results from information theory [11], the costs for encoding the model parameters is reliably estimated by:

$$paramCosts(c) = \frac{numParams(c)}{2} \cdot \log_2 |\mathcal{O}_c|$$

For any numerical cluster-specific attribute we need to encode its mean and variance while for a categorical one the probability deviations to the default concept hierarchy need to be encoded, i.e. $numParams(c) = |\mathcal{X}| \cdot 2 + \sum_{A_i \in \mathcal{A}} |N_c|$. Moreover, we need to encode the probabilities associated with the default concept hierarchy, as well as the default (global) means and variances for all numerical attributes. However, these costs are summarized to a constant term which does not influence our subspace selection and clustering technique.

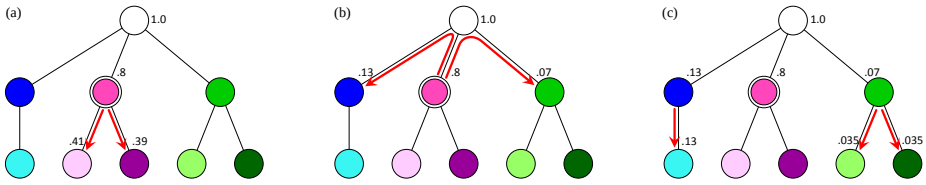

Fig. 2. Update concept hierarchies considering *pink* as a cluster-specific node.

3 Algorithm

Together with the main building blocks of ClicoT, two other steps are required to achieve an appropriate parameter free clustering: (1) recognizing the cluster-specific elements and (2) probability adjustments.

Cluster-specific Elements: Let the *specific coding cost* denote the cost where an element is marked as specific and the *non-specific coding cost* indicate the cost otherwise. Consulting the idea that cluster-specific elements have the most deviation of specific and non-specific cost and therefore saves more coding costs, we introduce a greedy method to recognize them. We iteratively sort the elements according to their deviations and specify the first element as a cluster-specific element. We continue marking elements until marking more elements does not pay off in terms of the coding cost. Note that different nodes of a concept hierarchy have the same opportunity to be specific.

Probability Adjustment: To adjust the probabilities for a numerical cluster-specific attribute we can safely use mean and variance corresponding to the cluster. In contrast, learning the cluster-specific concept hierarchy is more challenging since we need to maintain the integrity of a hierarchy. According to Definition 1 we assure that node probabilities of siblings in each level sum up to the probability of the parent node. Moreover node probabilities should sum up to one for each level. we provide a pseudocode concerning this procedure in appendix. To clarify, let Figure 2 show the procedure on the concept hierarchy corresponding to the running example (Figure 1) where labels denote the frequencies. Moreover, let pink be a cluster-specific node for the cluster with the shape $\times$. The adjustment starts with the root node and processes its children. Then it continues computing the relative probabilities for the specific concept hierarchy rather by background probability fraction (Figure 2a). 80% relative probability should be distributed between two children, rose and purple, based on the computed propagation factor. During the next step the remaining 20% probability is assigned level-wise to blue and green to assure that probabilities in each level sum up to 1 (Figure 2b). Again each parent propagates down its probability (Figure 2c). The result is a concept hierarchy best fitting to the objects when the background distributions are preserved.

ClicoT Algorithm: ClicoT is a top-down parameter-free clustering algorithm. That is, we start from a cluster consisting of all objects and iteratively split down the most expensive cluster c in terms of the coding cost to two new clusters $\{c'_a, c'_b\}$. Then, we apply a k-Means-like strategy and assign every point to closest cluster which is nothing else than the cluster with the lowest increase

Algorithm 1 ClicoT

input $\mathcal{DB}$
learn background distributions of each attribute
$C' = \{C_0\}$ with $C'_0 = O_i \in \mathcal{DB}$
repeat
  // try to split until convergence
  $C = C'$
  cost = $DL(\mathcal{DB}|C)$ // current cost
  $C' = \{C'_1 \dots C'_{k-1}\}$ split worst $C_i \in C$ to $\{C'_i, C'_k\}$
  while clustering C' changes **do**
    $C'_i = \{O_j : \min_i DL(O_j|C'_i)\}$ // assign objects
    Select cluster-specific elements by a greedy method for each cluster and compute costs
    Update each attribute of C'_i
  end while
  cost' = $DL(\mathcal{DB}|C')$ // split cost
until cost > cost'
$k = |C|$
return C, k

in the coding cost. Employing the greedy algorithm, we determine the cluster-specific elements and finally we compute the compression cost for clustering results in two cases, before and after splitting (Definition 1). If the compression cost after splitting, i.e. $\mathcal{C}'$ with $|\mathcal{C}'| = k + 1$, is cheaper than the cost of already accepted clustering $\mathcal{C}$ with $|\mathcal{C}| = k$ then we continue splitting the clusters. Otherwise the termination condition is reached and the algorithm will be stopped.

4 Related Work

Driven by the need of real applications, the topic of clustering mixed-type data represented by numerical and categorical attributes has attracted attentions, e.g. CFIKP [13], CAVE [7], CEBMDC [5]. In between, most of the algorithms are designed based on the algorithmic paradigm of k-Means, K-means-mixed (KMM) [1], k-Prototypes [8]. Often in this category not only the number of clusters k but also the weighting between numerical and categorical attributes in clustering has to be specified by the user. Among them, KMM avoids weighting parameters by an optimization scheme learning the relative importance of the single attributes during runtime, although it needs the number of clusters k as input parameter. Following a mixture of Gaussian distributions, model based clustering algorithms have been also proposed for mixed-type data. In between, clustMD [9] is developed using a latent variable model and employing an expectation maximisation (EM) algorithm to estimate the mixture model. However this algorithm has a certain Gaussian assumption which does not have to be necessarily fulfilled. Some of the approaches utiliz the unique characteristics of any data type to avoid the drawbacks of converting a data type to another

one. Profiting of the concept hierarchy, these algorithms introduce an integrative distance measure applicable for both numerical and categorical attributes. The algorithm DH [6] proposes a hierarchical clustering algorithm using a distance hierarchy which facilitates expressing the similarity between categorical and numerical values. As another method, MDBSCAN [2] employs a hierarchical distance measure to introduce a general integrative framework applicable for the algorithms which require a distance measure .e.g. DBSCAN. On the other hand, information-theoretic approaches have been proposed to avoid the difficulty of estimating input parameters. These algorithms regard the clustering as a data compression problem by hiering the Minimum Description Length (MDL). The cluster model of these algorithms comprises joint coding schemes supporting numerical and categorical data. The MDL principle allows balancing model complexity and goodness-of-fit. INCONCO [10] and Integrate [4] are two representative for mixed-type clustering algorithms in this family. While Integrate has been designed for general integrative clustering, INCONCO also supports detecting mixed-type attribute dependency patterns.

5 Evaluation

In this section we assess the performance of ClicoT comparing to other clustering algorithms in terms of NMI which is a common evaluation measure for clustering results. NMI numerically evaluates pairwise mutual information between ground truth and resulted clusters scaling between zero and one. We conducted several experiments evaluating ClicoT in comparison to KMM [1], INCONCO [10], DH [6], ClustMD [9], Integrate [4] and MDBSCAN [2]. In order to be fair in any experiment, we input the corresponding concept hierarchy to the algorithms which are not designed for dealing with it. That is, we encode the concept hierarchy as an extra attribute so that categorical values belonging to the same category have the same value in this extra attribute. Our algorithm is implemented in Java and the source code as well as the data sets are publicly available[5].

5.1 Mixed-type Clustering of Synthetic Data

In order to cover all aspects of ClicoT we first consider a synthetic data set. Then we continue experiments by comparing all algorithms in terms of the noise-robustness. Finally we will discuss the runtime efficiency.

Clustering Results: In this experiment we evaluate the performance of all the algorithms on the running example (Figure 1) while all parametric algorithms are set up with the right number of clusters. The data has two numerical attributes concerning the position of any data point and a categorical attribute showing the color of the points. Figure 3 shows the result of applying the algorithms where different clusters are illustrated by different colors. As it is explicitly shown in this figure ClicoT, with NMI 1, appropriately finds the initially

[5] https://bit.ly/2FkUB3Q

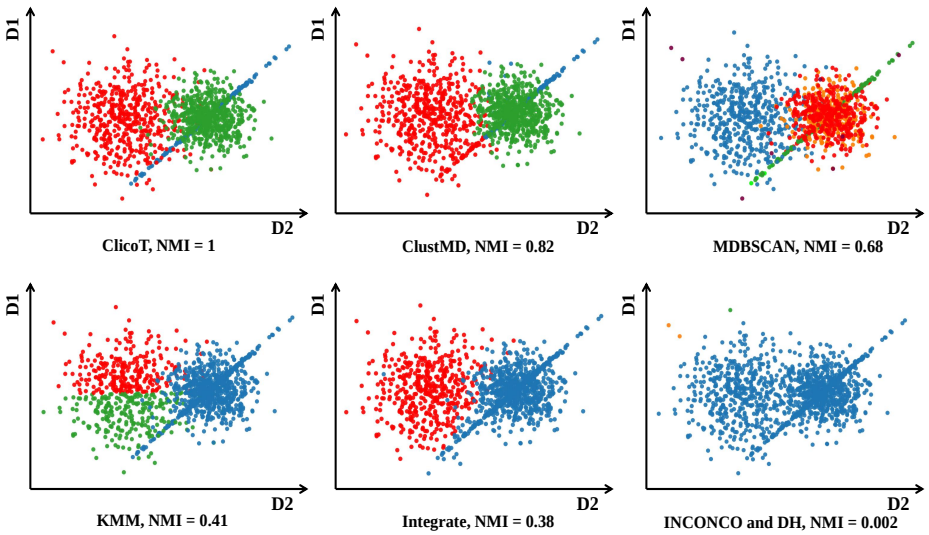

Fig. 3. Clustering results on the running example.

sampled three clusters where green, pink and blue are cluster-specific elements. Setting the correct number of cluster and trying various Gaussian mixture models, ClustMD results the next accurate clustering. Although MDBSCAN utilizes the distance hierarchy, but it is not able to capture the pink and green clusters. KMM can not distinguish among various colors. Since two clusters pink and green are heavily overlapped, Integrate can not distinguish among them. DH and INCONCO poorly result on this data set and they found almost only one cluster.

Noise-robustness: In this section we benchmark noise-robustness of ClicoT w.r.t the other algorithms in terms of NMI by increasing the noise factor. To address this issue we generate a data set with the same structure as the running example and we add another category, brown, to the categorical attribute color as noise. Regarding numerical attributes we increase the variance of any cluster. We start from 5 percent noise (noise factor = 1) and iteratively increase the noise factor ranging to 5. Figure 4 clearly illustrates noise-robustness of ClicoT comparing to others.

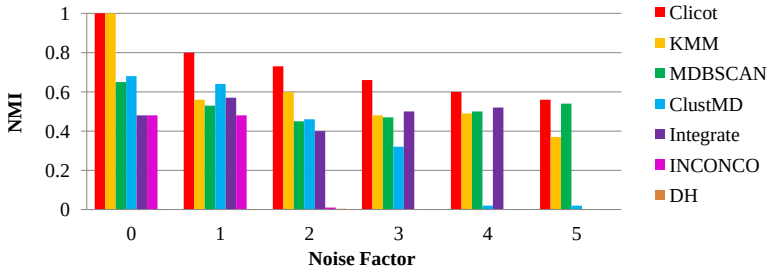

Fig. 4. Comparing noise-robustness of ClicoT to other algorithms.

Scalability: To evaluate the efficiency of ClicoT w.r.t the other algorithms, we generated a 10 dimensional data set (5 numerical and 5 categorical attributes)

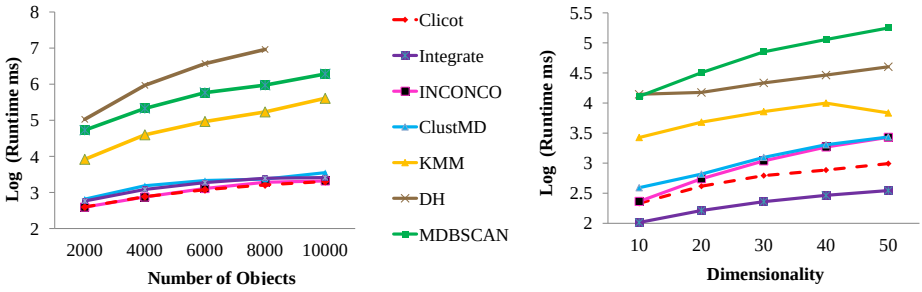

Fig. 5. Runtime experiment.

with three Gaussian clusters. Then respectively we increased the number of objects ranging from 2,000 to 10,000. In the other case we generated different data sets of various dimensionality ranging from 10 to 50 where the number of objects is fixed. Figure 5 depicts the efficiency of all algorithms in terms of the runtime complexity. Regarding the first experiment on the number of objects, ClicoT is slightly faster than others while increasing the dimensionality Integrate performs faster. However, the runtime of this algorithm highly depends on the number of clusters k initialized in the beginning (we set $k = 20$). That is, this algorithm tries a rang of k and outputs the best results. Therefore, by increasing k the runtime is also increasing.

5.2 Real Experiments

Finally, we evaluate clustering quality and interpretability of ClicoT on real world data sets. We used *MPG*, *Automobile* and *Adult* data sets from the UCI Repository as well as *Airport* data set from the public project *Open Flights*[6].

MPG: MPG is a slightly modified version of the data set provided in the StatLib library. The data concerns city-cycle fuel consumption in miles per gallon (MPG) in terms of 3 categorical and 5 numerical attributes consisting of different characteristics of 397 cars. We consider MPG ranging from 10 to 46.6 as the ground truth and divide the range to 7 intervals of the same length. The information about the concept hierarchy is provided in the appendix. Comparing ClicoT (NMI = 0.4) to the other algorithms INCONCO(0.17), KMM(0.37), DH(0.14), MDBSCAN(0.02), ClustMD(0.33) and Integrate(0). ClicoT correctly finds 7 clusters each of which compatible with one of the MPG groups. Cluster 2, for instance, is compatible with the first group of MPGs since the frequency of the first group in this cluster is 0.9. In this cluster American cars with the frequency of 1.0, cars with 8 cylinders with the frequency of 1 and model year in first group (70-74) with the frequency of 0.88 are selected as cluster-specific elements.

Automobile: This data set provides 205 instances with 26 categorical and numerical attributes. The first attribute defining the risk factor of an automobile has been used as class label. Altogether there are 6 different classes. Due to many missing values we used only 17 attributes. Comparing the best NMI

[6] http://openflights.org/data.html

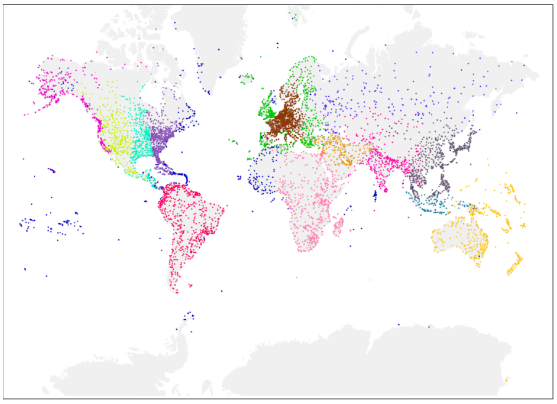

Fig. 6. Result of ClicoT on Open Flight data set.

captured by every algorithm, ClicoT (NMI = 0.38) outperforms kMM(0.23), INCONCO(0.20), Integrate(0.17), DH(0.04), ClusterMD(0.16) and MDBSCAN(0.02). Furthermore, ClicoT gives an insight in the interpretability of the clusters where Cluster 12, for instance, is characterized mostly by the fuel system of *2bbl*, but also by *1bbl* and *4bbl*. Also we see that Cluster 26 is consisting of both *mpfi* and slightly of *mfi*, too. Concerning the risk analysis this clustering serves, ClicoT allows to recognize which fuel systems share the same insurance risk.

Adult Data Set: Adult data set without missing values, extracted from the census bureau database, consists of 48,842 instances of 11 attributes. The class attribute Salary indicates whether the salary is over 50K or lower. Categorical attributes consist of different information e.g. work-class, education, occupation. A detailed concept hierarchy is provided in appendix. Although comparing to INCONCO(0.05), ClustMD(0.0003), MDBSCAN(0.004), DH(0) and Integrate(0), our algorithm ClicoT(0.15) outperforms all other algorithms except KMM(0.16) which is slightly better. But it seems that NMI does not sound a reasonable evaluation measure for this data set since there are only two classes in ground truth. ClicoT found 4 clusters in which Cluster 2, the biggest cluster consisting of almost 56% of objects, specifies *Husband* as the cluster-specific element, since it has the most deviation, but negative. The probability of instances having *Husband* as categorical value and the salary $<= 50K$ is zero in this cluster.Therefore along with the negative deviation this means that in Cluster 2 persons with the role as husband in a family earn more than $50K$.

Open Flights Data Set: The public project Open Flights provides world wide information about airports, flights and airlines. Here we consider instances of airports in order to carry out a cluster analysis. The data set consists of 8107 instances each of which represents an airport. The numeric attributes show the longitude and latitude , the sea height in meters and the time zone. Categorical attributes consist of the country, where the airport is located and the day light saving time. We constructed the concept hierarchy of the country attribute so that each country belongs to a continent. Since there is no ground truth provided for this data set we interpret the result of ClicoT (Figure 6) and we refer the reader to the appendix for more results regarding other algorithms.

Clustering results illustrated in Figure 6 consists of 15 clusters and shows that ClicoT appropriately grouped almost geographically similar regions in the

clusters. Starting from west to east, North American continent divided into five clusters. Obviously here the attribute of the time zone was chosen as specific because the clusters are uniquely made according to this attribute. Moving to the south, ClicoT pulled a plausible separation between South and North America. Considering South America as cluster-specific element and due to the rather low remaining airport density of South America ClicoT combined almost all of the airports to a cluster (red). In Western Europe there are some clusters, which can be distinguished by their geographic location. Additionally many airports around and in Germany are be grouped together.

6 Conclusion

To conclude, we have developed and demonstrated that ClicoT is not only able to cluster mixed-typed data in a noise-robust manner, but also yields most interpretable cluster descriptions. By using data compression as the general principle ClicoT automatically detects the number of clusters within any data set without any prior knowledge. Moreover, the experiments impressively demonstrated that clustering can greatly benefit from a concept hierarchy. Therefore, ClicoT excellently complements the approaches for mining mixed-type data.

References

1. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. **63** (November 2007)
2. Behzadi, S., Ibrahim, M.A., Plant, C.: Parameter free mixed-type density-based clustering. In: Database and Expert Systems Applications (DEXA) (2018)
3. Böhm, C., Faloutsos, C., Pan, J., Plant, C.: Robust information-theoretic clustering. In: KDD (2006)
4. Böhm, C., Goebl, S., Oswald, A., Plant, C., Plavinski, M., Wackersreuther, B.: Integrative parameter-free clustering of data with mixed type attributes. In: PAKDD (1). pp. 38–47 (2010)
5. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: A cluster ensemble approach. CoRR **abs/cs/0509011** (2005)
6. Hsu, C.C., Chen, C.L., Su, Y.W.: Hierarchical clustering of mixed data based on distance hierarchy. Information Sciences **177**(20), 4474–4492 (2007)
7. Hsu, C.C., Chen, Y.C.: Mining of mixed data with application to catalog marketing. Expert Syst. Appl. **32**(1), 12–23 (2007)
8. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. **2** (September 1998)
9. Mcparland, D., Gormley, I.C.: Model based clustering for mixed data: Clustmd. Adv. Data Anal. Classif. **10**(2) (2016)
10. Plant, C., Böhm, C.: Inconco: interpretable clustering of numerical and categorical objects. In: KDD. pp. 1127–1135 (2011)
11. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of Statistics **11**(2), 416–31 (1983)
12. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: ICML (2009)
13. Yin, J., Tan, Z.: Clustering mixed type attributes in large dataset. In: ISPA. pp. 655–661 (2005)