Dependency Clustering Across Measurement Scales

Claudia Plant
Florida State University
cplant@fsu.edu

ABSTRACT

How to automatically spot the major trends in large amounts of heterogeneous data? Clustering can help. However, most existing techniques suffer from one or more of the following drawbacks: 1) Many techniques support only one particular data type, most commonly numerical attributes. 2) Other techniques do not support attribute dependencies which are prevalent in real data. 3) Some approaches require input parameters which are difficult to estimate. 4) Most clustering approaches lack in interpretability. To address these challenges, we present the algorithm Scenic for dependency clustering across measurement scales. Our approach seamlessly integrates heterogenous data types measured at different scales, most importantly continuous numerical and discrete categorical data. Scenic clusters by arranging objects and attributes in a cluster-specific low-dimensional space. The embedding serves as a compact cluster model allowing to reconstruct the original heterogenous attributes with high accuracy. Thereby embedding reveals the major cluster-specific mixed-type attribute dependencies. Following the Minimum Description Length (MDL) principle, the cluster-specific embedding serves as a codebook for effective data compression. This compression-based view automatically balances goodness-of-fit and model complexity, making input parameters redundant. Finally, the embedding serves as a visualization enhancing the interpretability of the clustering result. Extensive experiments demonstrate the benefits of Scenic.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

clustering, heterogeneous data, Minimum Description Length

1. INTRODUCTION

In many applications, data are measured on different scales of measurement. For example in biomedicine, we often have binary attributes like sex and categorical attributes, like different genotypes. Moreover, we often have continuous valued attributes like laboratory parameters. To exploit the potential of the available information for knowledge discovery, we need data mining methods which support the integration of different data sources regardless of their measurement scale. As an example, each instance of the data set in Figure 1 is characterized by five attributes: The numerical attributes x and y and three categorical attributes: the attribute *color* with values *red, green* and *blue*; *symbol* with values *box* and *triangle*, and *filling* with values *open* and *filled*, respectively.

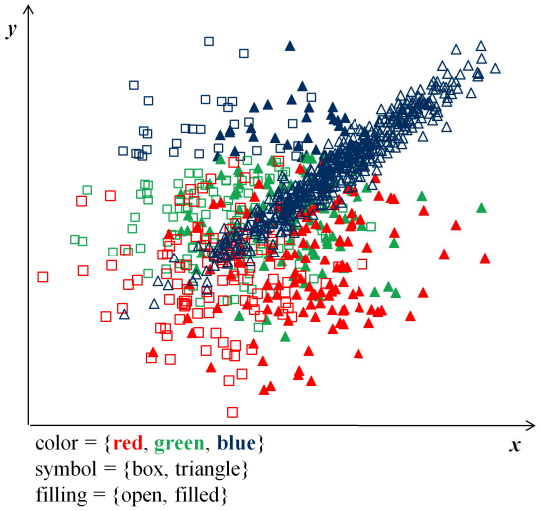
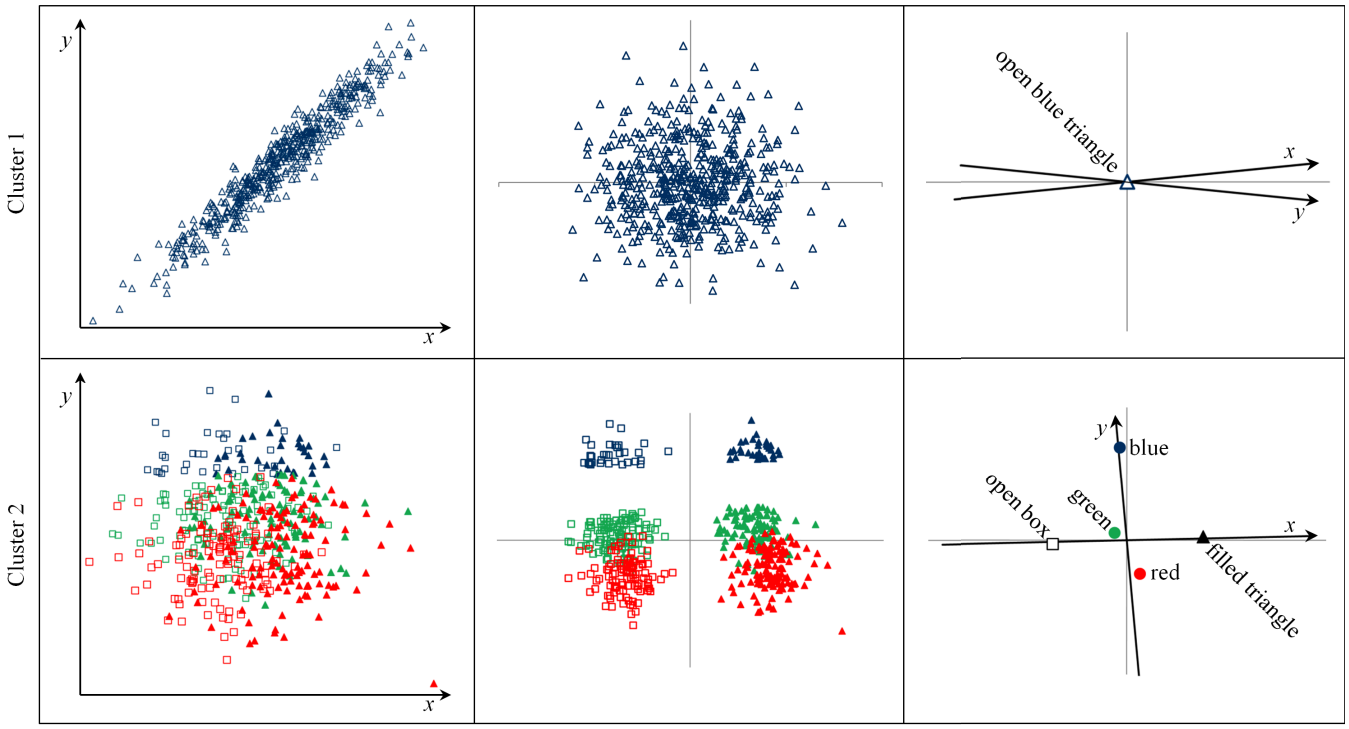


Figure 1: Running Example.

In many cases, the true intrinsic dimensionality of a mixed-type data set is much lower because of attribute dependencies. In complex data sets, attribute dependencies are often not global but exist at the level of single clusters. Our example consists of two clusters, see also Figure 2(a). In Cluster 1, we observe a strong dependency among the numerical x- and y-coordinates. In Cluster 2, there is no dependency among the numerical coordinates but two interesting mixed-type dependencies among numerical and categorical attributes: The attributes *symbol* and *filling* depend on the x value of the objects. For small x values, we mostly observe *open boxes*. The larger x the more likely we have *filled triangles*. Likewise, the attribute *color* depends on the y value: With increasing y we see the transition of colors from *red* over *green* to *blue*. In Cluster 2, the continuous attribute y basically measures the same information as the discrete attribute *color*, however on a different measurement scale. In many real applications, it is up to the expert to select a measurement scale, e.g. for a laboratory parameter either a continuous scale or just record the information whether or not the parameter is within some tolerance range. When integrating data from different sources, we can therefore expect to see a

KDD'12, August 12–16, 2012, Beijing, China.



(a) Original Space (b) Embedded Objects (c) Embedded Attributes

Figure 2: Single Clusters of Running Example: (a) Original space; (b,c) objects and attributes in Attribute-object (AO) space.

wide range of mixed-type attribute dependencies which are very interesting for interpretation. Considering attribute dependencies regardless of the measurement scale opens up novel opportunities in clustering heterogeneous data. In clustering numerical data, *correlation clustering* has attracted much attention. A large volume of research papers, such as the approaches ORCLUS [1], 4C [4] and CURLER [14] have demonstrated the potential of integrating Principal Component Analysis (PCA) into clustering. These techniques detect clusters in arbitrarily oriented subspaces corresponding to unique correlation patterns revealed by PCA. The cluster-specific subspaces are very useful for interpretation since they explain why objects are clustered together. However, all these approaches are suitable for vector data only, i.e. for data of continuous scale level. We therefore introduce a novel cluster notion for mixed-type data.

1.1 Basic Idea

A cluster is a set of objects characterized by a unique attribute dependency pattern. We detect and resolve mixed-type attribute dependency patterns by embedding the data objects and the attributes into a joint low-dimensional vector space, which we call the Attribute-object (AO) space. Attributes and objects are arranged such that the low-dimensional distances between the objects and their corresponding attribute values are minimized. This type of embedding has been proposed in [11], but has so far not been combined with or integrated into clustering. To respect the order and spacing constraints in numerical data, continuous attributes are represented as lines in low-dimensional space and for nominal attributes, every category is embedded. Figures 2(b) and 2(c) display the embedded objects and attributes of the two clusters of our running example. For comparison, Figure 2(a) displays the single clusters in original space. For clarity of presentation, we display the embedding of the objects and of the attributes in separate sub-figures but note that actually all objects and attributes lie in a common vector space. For each cluster a separate embedding is generated to respect the cluster-specific attribute dependencies. The embedding of the attributes in Figure 2(c) reveals at first glance the major characteristics and dependencies of a cluster. Cluster 1 is composed of *open blue triangles*. Therefore these three categories are placed at the center of the display. We can see that both attributes x and y measure similar information, since there is a small angle between the corresponding lines in AO space. The embedding de-correlates the data similar as PCA, c.f. Figure 2(b). Cluster 2 is composed of *open boxes* and *filled triangles* of all colors. The complex mixed-type dependency pattern in Cluster 2 becomes obvious and accessible for interpretation from Figure 2(c). We can clearly see the strong dependency between attribute x and the attributes *filling* and *symbol*. Likewise, we see that the *color* value depends on the y-coordinate. We can even see that the *blue* objects are clearly separated from the remaining colors by the y value, whereas *red* and *green* objects overlap.

Our novel objective function directly relates clustering 1) to unsupervised classification and 2) to data compression. Since each cluster represents a unique attribute dependency pattern, a unique vector space representation of the original attributes is an essential part of the cluster model. We exploit this cluster-specific low-dimensional embedding to reconstruct the original attribute values with high accuracy. Therefore, we search for clusters which predict or explain the original mixed-type attribute values in the data with high accuracy. However, the prediction accuracy alone is not sufficient to comprehensively assess the quality of a clustering result: The more clusters and the more complex the cluster models, i.e. the higher dimensional the vector space representation of each cluster model, the better is the prediction accuracy. To trade off goodness-of-fit and model complexity, we therefore combine the idea of unsupervised classification with the idea of data compression.

1.2 Contributions

The benefits of our approach can be summarized as follows:

- We introduce a novel **cluster notion to support generalized dependency clustering of data measured at different measurement scales**.
- The novel **algorithm Scenic** (Scale-free Dependency Clustering) clusters by embedding objects and attributes in a joint low-dimensional vector space which is very useful for **interpretation** of the result.
- Linking clustering to unsupervised classification and data compression, Scenic produces **valid results without overfitting**.

Notation. In the following we consider a data set DS with n objects. Each object x is represented by d attributes. Attributes are denoted by capital letters and can be either numerical features or categorical variables with two or more values. We denote the number of categorical attributes by d_c and the number of numerical attributes by d_n. For a an attribute A, we denote a value (category or numerical value) by a. We further denote by a_x the value of object x and attribute A. The representation of an object x in AO space is denoted by $\pi(x)$, and the representation of a value a of attribute A by $\pi(a)$. The number of categories of A is denoted by $|A|$. The result of our algorithm is a disjoint partitioning of DS into k clusters $\mathcal{C}_1, ..., \mathcal{C}_k$. The remainder of this paper is organized as follows: In the next section, we elaborate our novel cluster notion and the clustering objective. In Section 3, we introduce the algorithm Scenic. Section 4 is dedicated to an extensive experimental evaluation. Section 5 surveys related work and Section 6 concludes the paper.

2. CLUSTER NOTION AND CLUSTERING OBJECTIVE

In this section, we introduce our cluster model based the Attribute-object space, a low-dimensional vector space representation for mixed-type data suitable for unsupervised classification and data compression. We start by formally introducing the AO space and illustrating its basic properties.

2.1 Basic Properties of the AO Space

DEFINITION 1 (ATTRIBUTE-OBJECT SPACE.). *The AO space of dimensionality d_v of a cluster consists of the following:*

- *The $n \times d_v$ matrix $\Pi(x)$ containing the low-dimensional* object coordinates *$\pi(x)$ of each object x as row-vectors;*
- *the $c \times d_v$ matrix $\Pi(a)$ containing the low-dimensional* attribute coordinates *$\pi(a)$ for each attribute A and each value a, where c denotes the total number of distinguishable attribute values of all numerical and categorical attributes in the data.*

Each numerical attribute A imposes order and spacing constraints on $\pi(a)$:

- Order constraint: *For every pair of values a_1 and a_2, every numerical attribute A and every dimension d of the AO space holds: $a_1 < a_2 \Rightarrow \pi(a_1)_d < \pi(a_2)_d$.*
- Spacing constraint: *For every three values a_1, a_2, a_3 holds: $a_1 = a_2 \cdot a_3 \Rightarrow \pi(a_1)_d = \pi(a_2)_d \cdot \pi(a_3)_d$.*

The goal of the AO space is to represent the major aspects of the complex high-dimensional similarity among mixed-type data objects and attributes in a compact form. Similar as Principle Component Analysis, the transformation to AO space distills the major characteristics from the data by 1) resolving attribute dependencies and 2) removing noise. To learn a suitable AO space, it is essential to respect the fundamental difference in nature between nominal and continuous attributes. The order and spacing constraints guarantee that continuous attributes appear as lines in AO space. Actually, each single value is embedded, but all values lie on a line which we therefore display continuous instead of dotted in Figure 2(c). Only by this restriction, we obtain simple and thus interpretable mixed-type dependencies such that with increasing x we observe more likely *filled triangles* than *open boxes*. Nominal attributes have no order or spacing on their categories and therefore their location in AO space is not restricted. It would be even very counterproductive to restrict their location since by embedding we want to learn their hidden order and spacing from the overall data. It is also not necessary to restrict the location of the objects in any way. Considering Figure 2(b), the objects are clearly not arbitrarily embedded. However, it is important to note that any arbitrary embedding of the objects and the nominal categories would be a valid AO space as long as the order and spacing constraints on the numerical attributes are respected.

2.2 Unsupervised Classification and Data Compression

Definition 1 specifies what a valid AO space is. But what is a good AO space? It is essential to answer this question for the following reasons: Our example demonstrates that especially the embedding of the attributes is very helpful to interpret the cluster content (cf. Figure 2(c)), however we must clarify how much we can trust what we see. We also need a quality measure to compare different AO spaces which are produced as intermediate results of our algorithm, e.g. two sets of AO spaces resulting from different partitioning of the objects into clusters.

A key idea of this paper is to regard clustering as an unsupervised classification problem in a novel way: Given the low-dimensional AO representation of objects and attributes, we want to predict the original attribute values (i.e. the mixed-type input data) with high accuracy. A cluster corresponds to an unknown class of the data. The AO space of a cluster is a specific classification system accustomed to the data distribution in the cluster. The goal of clustering is to find a grouping of objects which maximizes the prediction accuracy.

However, the idea of unsupervised classification alone is not a suitable optimization goal. To see this, consider a clustering where each object is placed in its own singleton cluster. For each singleton cluster, the embedding AO space always provides optimal prediction accuracy. Considering the nominal attributes, each AO space would only contain those categories which the corresponding object has. For the numerical attributes all lines would shrink to points. Thus, each object would be embedded together with its individual attribute values only. No matter what classification scheme we would apply, we could perfectly predict the original values since there are no incorrect answers in the solution space. However, such clustering result would be very bad for interpretation since we learn nothing about the data. Therefore we must take action against overfitting.

To avoid overfitting we combine the idea of unsupervised classification with the idea of data compression, also known as the Minimum Description Length (MDL) Principle. Suppose, we want to transfer the data via a communication channel from a sender to a receiver. A good AO space is a compact model of the data which can drastically reduce the communication costs. The description length DL of a cluster $\mathcal{C}_i$ corresponds to:

$$DL(\mathcal{C}_i) = RE_{\mathcal{C}_i} + MC_{\mathcal{C}_i}.$$

The reconstruction error $RE_{\mathcal{C}_i}$ denotes the number of bits required to reconstruct the original feature information, i.e. the numerical and categorical attribute values of all objects using the AO space as a model. The model complexity $MC_{\mathcal{C}_i}$ corresponds to the number of bits required to encode the AO model itself and includes the bits required to encode the low-dimensional object coordinates as well as the model parameters.

2.2.1 Reconstruction Error

Categorical Attributes. Considering a single object x and categorical attribute A, we determine the reconstruction error $RE_{x,A}$ using Bayes's theorem and Huffman Coding: For each category a of A, the probability to observe a given the representation $\pi(x)$ of x in AOL space is provided by:

$$p(a|\pi(x)) = \frac{p(a) \cdot p(\pi(x)|a)}{p(\pi(x))}.$$

With a suitable model for the probability distribution of category a in AO space, we can specify the probability to observe $\pi(x)$ given that category a. From Section 3.1 it will be evident that applying a Gaussian PDF $\mathcal{N}_a(\mu_a, \Sigma_a)$ with diagonal covariance is an appropriate choice. Thus,

$$p(\pi(x)|a) = (2\pi)^{-\frac{d_v}{2}} \cdot |\Sigma_a|^{-\frac{1}{2}} \exp^{(\pi(x)-\mu_a)^t \cdot \Sigma_a^{-1} \cdot (\pi(x)-\mu_a).}$$

Applying additionally $p(a) = \frac{|a|}{n}$ and $p(\pi(x)) = \sum_{a \in A} p(\pi(x)|a)$ we can deduce the probability of category a given the AO representation $\pi(x)$. Informally, a good AO space predicts the true category of an object x with high accuracy. This implies, in a good AO space we obtain a very high probability, ideally almost 1, for $p(a_x|\pi(x))$, denoting by a_x the true category of object x, and a very low probability to all remaining categories of attribute A. The reconstruction error of the AO space corresponds to the amount of uncertainty on the categorical attribute values which remains after embedding. To quantify the reconstruction error in bits, we use the principle of Huffman coding summing up over all objects in a cluster $\mathcal{C}_i$ and all categorical attributes:

$$RE_{cat} = \sum_{x \in \mathcal{C}_i} \sum_{A_{cat}} -\log_2(p(a_x|\pi(x))).$$

Numerical Attributes. To quantify the reconstruction error RE_B of a numerical attribute B for the objects of a cluster $\mathcal{C}_i$, we apply multivariate regression where the AO coordinates of the objects $\pi(\mathcal{C}_i)$ are the regressors and the original attribute values $\mathcal{C}_{i,B}$ correspond to the dependent variable:

$$\mathcal{C}_{i,B} = \pi(\mathcal{C}_i) \cdot \beta_B + \epsilon_B.$$

This is a standard linear regression with a Gaussian error distribution. The coding costs therefore correspond to the negative log-likelihood of the error distribution summed up over all objects and attributes:

$$RE_{num} = \sum_{x \in \mathcal{C}_i} \sum_{A_{num}} -\log_2\left(\frac{1}{\sigma_{\epsilon_A}\sqrt{2\pi}} \exp^{-\frac{(x-\mu_{\epsilon_A})^2}{2\sigma^2_{\epsilon_A}}}\right).$$

The overall reconstruction error RE is provided by summing up the numerical and categorical reconstruction errors:
$RE_{\mathcal{C}_i} = RE_{num} + RE_{cat}$.

2.2.2 Model complexity

Following [13], we estimate the costs to encode the model parameters by:

$$P_{cost} = \frac{|m|}{2} \cdot \log_2 |\mathcal{C}_i|,$$

where $|m|$ stands for the number of model parameters. The number of model parameters $|m| = d_v \cdot n + 2 \cdot d_v \cdot |cat| + d_v \cdot d_n$, where the first term corresponds to the costs required to encode the object representations, the second term to the category means and variances and the third term is required for the linear models of the numerical attributes, i.e. the model coefficients β. Besides the model complexity of the cluster-specific embeddings, we need to specify the cluster identifier for each object. Using Huffman coding, these id-costs account for $ID_{cost} = |\mathcal{C}_i| \cdot \log_2(\frac{n}{|\mathcal{C}_i|})$. The overall model complexity is the sum of the parameter costs and the id-costs:

$$MC_{\mathcal{C}_i} = P_{cost} + ID_{cost}.$$

2.2.3 Clustering Objective

The clustering objective is to find a partitioning of the objects into clusters such that the overall description length is minimized:

$$\min \sum_{\mathcal{C}_i \in \mathcal{C}} DL(\mathcal{C}_i).$$

It is important to note that the number of clusters is not an input parameter but is determined by our algorithm at runtime guided by the description length. Likewise, the description length allows us to select a suitable dimensionality of the individual cluster-specific embeddings.

3. ALGORITHM SCENIC

Having specified the cluster notion and clustering objective we now introduce an efficient and effective algorithm to find a good clustering. In particular, our algorithm needs to give answers to the following two questions: How to find a good AO space for a cluster? And: How to find the clusters? This section discusses the answers of our approach Scenic.

3.1 Detecting a Suitable AO Space

Intuitively, it is clear that in an AO space suitable for unsupervised classification and data compression, objects must be embedded as close as possible to the embeddings of their own correct attribute values. At the same time, objects should be as far away as possible from the embeddings of the attribute values they do not have. For example, an object having the value *blue* at the categorical attribute *color* should be embedded as close as possible to the embedding of the category *blue* and at the same time as far away as possible from the values *red* and *green* in order to guarantee correct prediction and effective compression, cf. Section 2.2.1.

Interestingly, the claim that the object should be placed as close as possible to its correct values and as far as possible to the values of other objects resembles a lot to clustering. In clustering, like K-means we claim that objects in a common cluster should be as similar as possible and objects in different clusters should differ as much as possible. As building block to detect a suitable AO space we use the algorithm Princals [11], which is, similar as K-means, an alternating least squares algorithm. Princals first initializes the object coordinates randomly. To avoid the trivial solution that all objects are embedded to the same location, the random coordinates are column-centered and orthogonalized. Princals then iterates two steps until convergence: In step 1), coordinates for the attribute values are determined as the mean of all assigned objects, i.e. all objects having this value. To preserve the order and spacing constraints, the location of the numerical attribute values is corrected by a linear regression using the original attribute values as regressors. In step 2), coordinates for the objects are determined by the mean of all category locations to which the object belongs. After the second step, the object coordinates are orthogonalized, e.g. by using the Gram-Schmidt procedure. When the data is solely numerical, Princals yields the same result as PCA of the centered and normalized data.

```
algorithm Scenic (): set of clusters
  Cluster C_S := k-Scenic (1);
  return REC-SPLIT (C_R) ;
  determine best AO of clusters with MDL ;

algorithm k-Scenic (k): set of k clusters
  {C_1, ..., C_k} := INITIALIZATION (k);
  repeat
    assign every object to C_i with minimum coding cost;
    update AO space as in Section 3.1;
  until convergence;
  return {C_1, ..., C_k};

procedure INITIALIZATION (k): set of k clusters
  k-d Princals;
  obtain l object clusters for each category combination;
  while l > k do
    merge most similar category combination;
  return {C_1, ..., C_k};

procedure REC-SPLIT (Cluster C): set of clusters
  {C_L, C_R} := k-Scenic (2);
  if MDL(C_L) + MDL(C_R) ≥ MDL(C) then
    return {C};
  else
    return REC-SPLIT (C_L)∪ REC-SPLIT (C_R);
```

Figure 3: The Algorithm Scenic.

Applying a Gaussian PDF in Section 2 is justified by the fact, that Princals embeds each category a at the center of all objects having the value a. Therefore, all category centroids also share a common variance. Moreover, it is appropriate to apply a spherical Gaussian without covariance matrix and to consider each dimension separately because of the orthogonality of the AO dimensions.

3.2 Finding the Clusters

We start by introducing the algorithm k-Scenic for dependency clustering with a fixed number of clusters k and finally introduce Scenic which combines k-Scenic with an effective top-down splitting strategy for parameter-free clustering. Figure 3 summarizes the algorithm in pseudocode.

Since k-Scenic is a k-means-style algorithm which is initialization dependent, we propose the following initialization strategy. Inspired by spectral clustering [10], k clusters can be well separated in a k-dimensional AO space. Similar to PCA, the k first dimensions of the AO space can be regarded as the major dimensions explaining most of the variance. Due to the special properties of the AO space and the algorithmic scheme of Princals, the objects of each category combination form a Gaussian in AO space. If we have more than k category combinations, we greedily merge in each step the most similar category combinations until we end up with k clusters. We merge two category combinations by replacing their multivariate Gaussians with the representative having minimal Kulback-Leibler divergence to both of them. The parameters of this representative can be determined efficiently in closed form, cf. [5].

After initialization, k-Scenic iterates two phases until convergence: 1) Assignment of each object to that cluster where it has the minimal coding costs, and 2) update of the cluster model which involves updating the AO space. Starting with all objects in one cluster, the algorithm Scenic recursively applies 2-Scenic as long we observe an improvement in coding costs. During the splitting phase, clustering is performed in 2-dimensional AO space. In the end, the best AO dimensionality of each individual cluster is determined with MDL.

The runtime complexity of Scenic depends on the number of iterations within the k-Scenic invocations and the number of iterations within Princals. The runtime for k-Scenic is $n \cdot |iter_{kSc}| \cdot (|iter_P| \cdot nd_v^2)$, where the last term represents the time needed for Gram-Schmidt orthogonalization. Since the number of iterations in k-Scenic $|iter_{kSc}|$ and the number of iterations in Princals $|iter_P|$ usually is small (about 10 - 50), the algorithm is efficient.

4. EXPERIMENTS

In this section we perform experiments comparing Scenic to INCONCO [12] and K-Means Mixed [2], two state-of-the-art techniques for clustering mixed-type numerical and categorical data. As a baseline, we also compare to K-means and K-modes. As clustering quality measure we report the Normalized Mutual Information (NMI) [15]. This score scales between 0 and 1. The higher the NMI the better is the clustering.

4.1 Synthetic Data

Figure 4 displays the clusters found by Scenic (a) and the comparison methods (b-d) on our running example. The data set consists of 1,000 objects represented by three nominal and two numerical attributes. Each of the two clusters is composed of 500 objects exhibiting a cluster-specific attribute dependency pattern, see also Figure 1 for the complete data set. In both clusters, the numerical coordinates exhibit a Gaussian distribution, spherical in Cluster 2 and with strong covariance in Cluster 1 (explaining 80% of the variance). Cluster 1 consists of *open blue triangles*. The categorical values of Cluster 2 have been assigned as follows: With probability $p := CDF(x_i)$ of the Gaussian cumulative density function at the x-coordinate of object i we observe *bold triangles*, with (1-p) probability *open boxes*. Analogously for the colors *red* and *green* and the y-coordinate; additionally the color *blue* is assigned for objects with $CDF(y_i) \geq 0.9$.

Scenic is the only method perfectly clustering this data set labeling the objects exactly to the ground truth as generated. Scenic successfully detects the numerical correlation in Cluster 1 and the mixed-type dependency in Cluster 2 and therefore achieves a perfect NMI of 1.0. The embedding of the attributes and the objects is a valuable source of information for interpreting the result. Especially the embedding of the attributes displayed in Figure 2(c) clearly visualizes the core characteristics of the clusters and the attribute dependencies: Categories placed at the center of the display like *open, blue, triangle* in Cluster 1 represent the typical attribute values of a cluster. Analogously, we can see that for Cluster 2, *green* and *red* objects are more frequent than *blue* ones, and the cluster contains *open boxes* as well as *filled triangles*. The complex mixed-type dependency pattern also becomes obvious at first glance: We have a correspondence between numerical x and the probability of observing *open boxes* or *filled triangles* which is displayed in the figure by the placement of those categories at the x-axis; analogously for the colors and the y-axis. Scenic selects to embed Cluster 1 in one-dimensional space and Cluster 2 in two-dimensional space.

The second best result with an NMI of 0.71 is obtained by K-means Mixed [2], cf. Figure 4(b). This clustering is guided by the attribute *color*: All *blue* objects are assigned to Cluster 1. Cluster 2 consists of the remaining *red* and *green* objects. K-means Mixed employs an optimization scheme to decide upon the relative importance of the single attributes for clustering. Among them, *color* best distinguishes between the clusters, since there are only relatively few *blue* objects in Cluster 2. However, only a technique considering dependencies between attributes of arbitrary scale can successfully cluster this data: The *blue* objects wrongly assigned to Cluster 1 by K-means-Mixed do not fit at all into this cluster since they have no correlation among the numerical x- and y-coordinates. How-

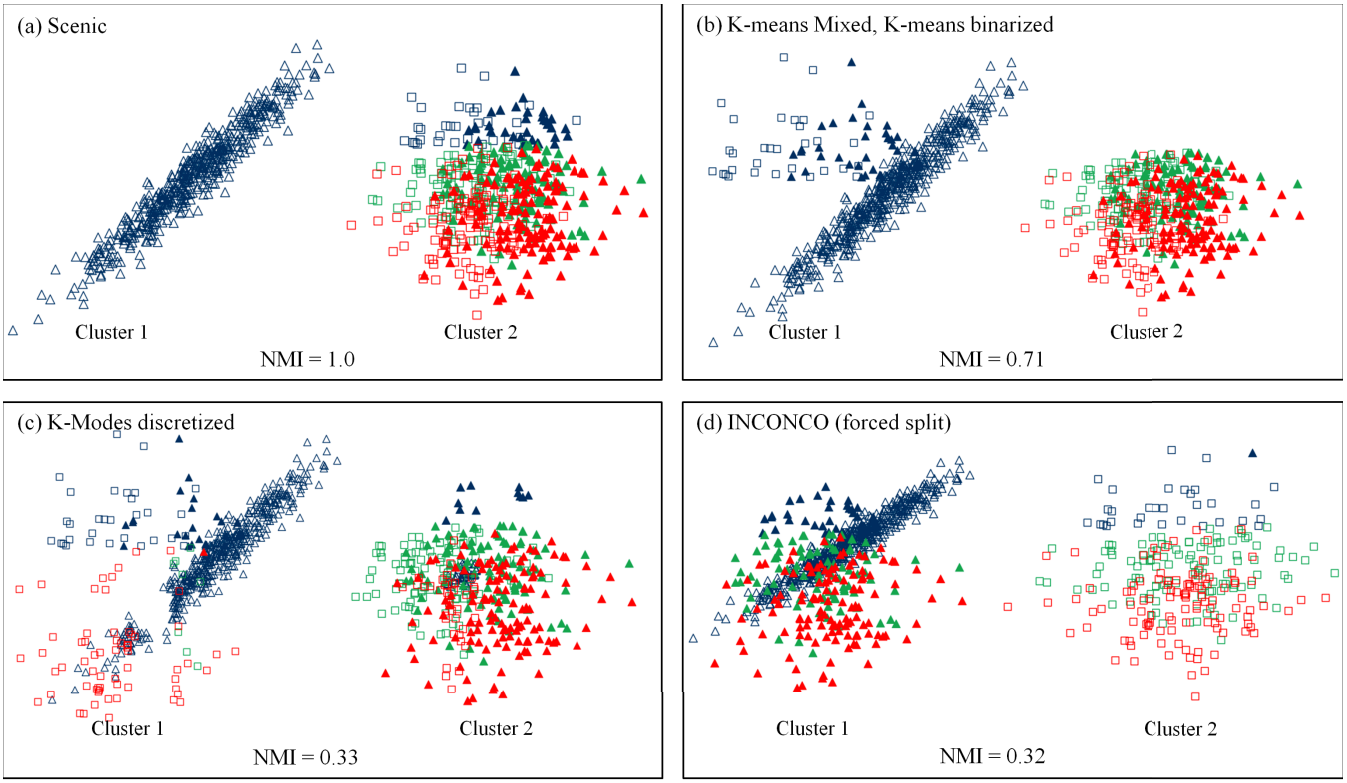


Figure 4: Results on Running Example: single clusters found by Scenic (a) and the comparison methods (b-d). Please also refer to Figure 1 for the complete data set and to Figure 2 for the embedding of objects and attributes performed by Scenic.

ever, these objects fit very well to Cluster 2 because of the transition from *open boxes* to *filled triangles* for increasing x-coordinate. The same result as with K-means Mixed can be obtained with standard K-means when we binarize the categorical attributes. Using solely the two numerical attributes in standard K-means yields a poor result with NMI of 0.18. It is evident that the categorical attributes contain important information for cluster separation, thus integrating both types of attributes improves the clustering result.

Another possibility is using the algorithm K-modes for categorical data and integrating the numerical information by discretizing the numerical attributes. This option yields an NMI of 0.33 using 10 equidistant bins, cf. Figure 4(c). This clustering is mainly guided by the attribute *filling*: Cluster 1 contains in majority *open* objects and Cluster 2 *filled* ones. Since the number of bins is difficult to select we report the result with the best NMI among several trials. Integrating the other source of information by type conversion is not an ideal solution but also helps in this case. Running K-modes solely on the nominal data yields an NMI of only 0.27.

Figure 4(d) shows the result of INCONCO [12]. This algorithm is in principle capable to consider mixed-type attribute dependency patterns, however has a special and limited dependency model: In a mixed-type attribute dependency, all categories of the involved nominal variables must have a unique numerical data distribution which is modeled by a separate Gaussian for each single category. In the cluster model of INCONCO it is not possible that only some of the categories are involved in a dependency. For example in Cluster 2, we observe a transition from *open boxes* to *filled triangles* for increasing x value. This dependency involves the attributes *symbol* and *filling*, but only one of the two categories of each attribute. INCONCO would model this dependency by having an own Gaussian not only for *open boxes* and *filled triangles* but also for *filled boxes* and *open triangles*, attribute combinations which are not existing in this cluster. In addition, the assumption of having a unique Gaussian for each category combination does not fit to the data distribution. As Scenic relying on the MDL, INCONCO selects the number of clusters automatically. Since the data does not fit to the model assumptions of the algorithm, INCONCO prefers to keep all objects in one cluster, a solution having an NMI of 0. To enable comparison, we forced a split into two clusters. The resulting clustering with an NMI of 0.32 is mostly distinguishing *open boxes* (Cluster 2) from the remaining objects (Cluster 1).

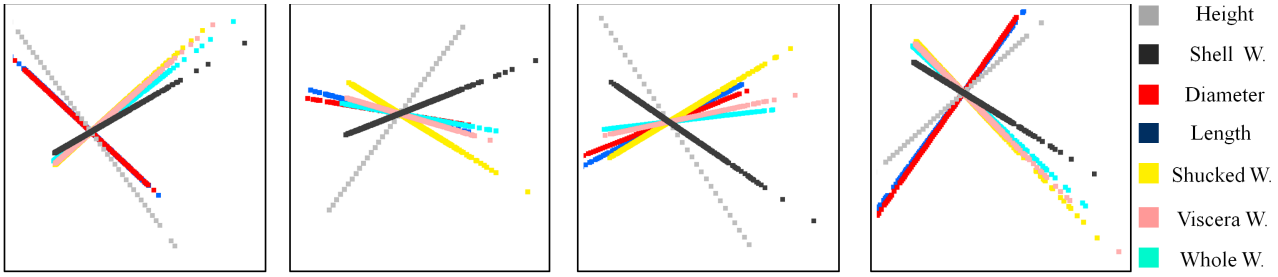
Runtime. Scenic is faster than K-means Mixed. However, INCONCO and of course also K-means and K-modes are faster than Scenic. For example to process a data set with 5,000 points and data distribution as the running example, Scenic needs 28 seconds, K-means Mixed 134 seconds, INCONCO, K-modes and K-means less than a second.

4.2 Real Data

We compare to K-means Mixed [2] and INCONCO [12] on two real data sets available at the UCI Machine Learning repository [6]. K-means and K-modes have been left out due to worse results.

4.2.1 Abalone

The abalone data set consists of 4,177 instances which are described by nine attributes: The categorical attribute *sex* with three values *male, female* and *infant*. Further eight numerical attributes represent several measurements of the abalone shell, cf. Figure 5; finally the integer-valued attribute *rings* providing the number of rings of the shell which allows inferring the age of the animal. The data has been originally collected to study the population biology of abalone in Tasmania in an unsupervised way. Since added to the UCI Machine Learning repository [6], the attribute *rings* has been mostly used for evaluation purpose. Therefore, we also excluded



(a) Cl. 1, Infant (b) Cl. 2, Female (c) Cl. 3, Mature Male (d) Cl. 4, Young Male

Figure 5: Cluster-specific Embedding of Attributes on Abalone Data.

this attribute from clustering and used it for evaluation. Scenic detects four clusters on this data set which correspond well to the attribute *sex*. Cluster 1 is purely composed of the 1,342 *infants* and Cluster 2 consists all 1,307 *females*. The *males* are split up into two clusters because Scenic identifies two groups of *male* instances having different attribute dependency patterns. Figure 5 displays the embedding of the attributes for all four clusters. For clarity of presentation, we omitted the attribute *sex*. Since all clusters are *sex*-pure, the corresponding value is embedded at the center of all displays. Note that different rotation in the embedding has no semantic meaning. From Figure 5 it is obvious that we have two general types of dependencies in the data: The young *males* and the *infants* are very similar (Cluster 1, displayed in Figure 5(a) and Cluster 4, cf. Figure 5 (d)). In both clusters, the attributes *length* and *diameter* are strongly correlated. The attributes *shucked weight, viscera weight and whole weight* are also correlated but represent different information. *Height* is rather correlated with *length* and *diameter*; *shell weight* with the other weight attributes. In mature *males* and *females*, cf. Figure 5(b) and 5(c), all attributes with exception of *height* and *shell weight* are rather correlated.

It is probably not surprising that the grouping into *sex*-pure clusters performed by Scenic is informative regarding the evaluation attribute *rings*: Especially the *infant* cluster 1 has with 7.89 a much smaller average number of *rings* than the other clusters (Cluster 2 (*females*): 11.13, Cluster 3 (*males*): 10.95, Cluster 4 (*males*): 10.37). Running a two-sample t-test on the *ring* distribution of each pair of clusters assuming unequal variance provides further evidence that it is reasonable 1) to cluster this data set according to the *sex*, and 2) to split up the *males* into two subgroups. Corrected with Bonferroni for multiple comparisons, almost all pairs of clusters exhibit significant differences in the number of *rings* (at level $\alpha = 0.05$). In particular, Cluster 3 and 4 differ significantly with p-value of 0.001 (corrected). Thus, as already observed above, Cluster 3 represents the mature *males* and Cluster 4 represents the younger *males* having attribute dependencies similar to the *infants*. The only pair of clusters not exhibiting a significant difference in age are Cluster 3 representing the mature *males* and Cluster 4 representing the *females*.

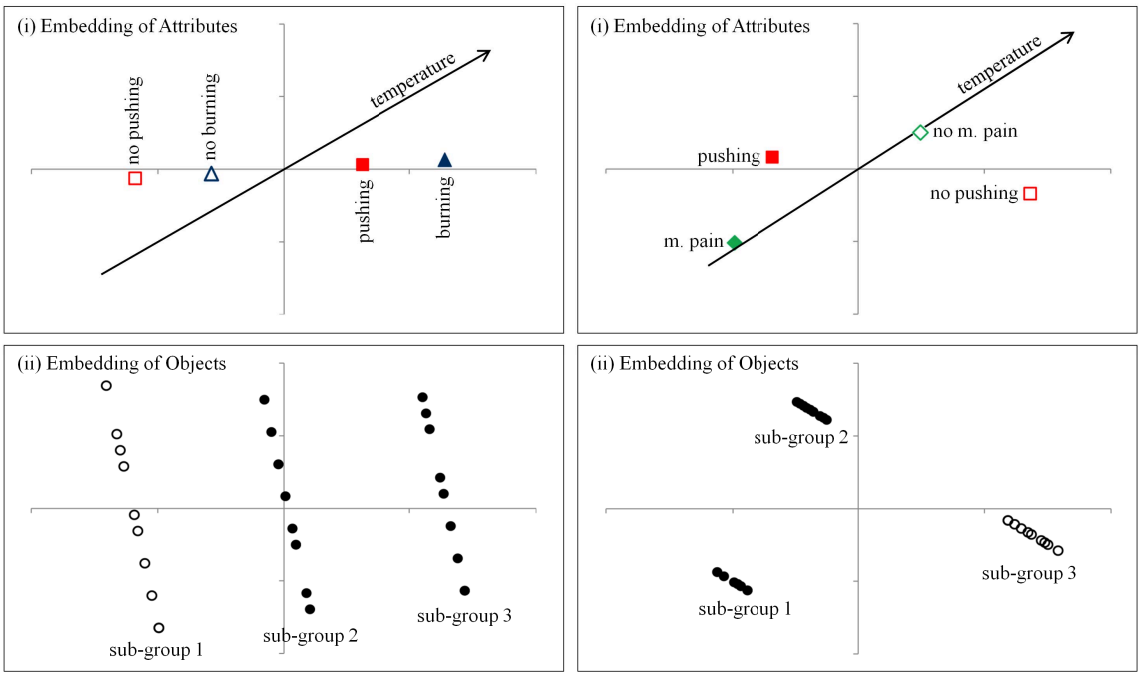
For comparison, we parameterized K-means Mixed to also detect four clusters. The result mainly separates the *infants* from the rest of the data. Cluster 1 consists of 1,337 *infants*. The other three clusters contain *males*, *females* and the few remaining *infants* to almost equal proportions. Regarding the number of *rings*, the result of K-means mixed is also reasonable since all pairs of clusters differ significantly in a two-sample t-test on this attribute. However, since K-means Mixed does not support attribute dependencies and the only output is the grouping of objects into clusters, the result is difficult to interpret: We do not know why objects are grouped together and which attributes are important for clustering.

INCONCO detects 25 clusters on this data set. The four largest clusters are consist of more than 300 objects. None of these clusters represent a particular *sex*, e.g. the largest cluster with 1,056 instances consists of 554 *males*, 442 *females* and 60 *infants*. Also regarding the number of *rings*, the result is not specific. None of the clusters stands out significantly. Since INCONCO lacks a visual representation of the attribute dependency patterns and detects many small clusters, the detected dependencies are hard to interpret and to compare across clusters. In summary, Scenic achieves all we want: 1) Scenic automatically selects a meaningful number of clusters. 2) The embedding of the attributes clearly visualizes the most important attributes and dependency patterns for clustering. 3) The clusters differ significantly in the evaluation attribute.

4.2.2 *Acute Inflammations Data*

This data set consists of several measurements relevant for the diagnosis of acute inflammation of the urinary bladder and acute nephritis. Each of the 120 instances represents a patient and is characterized by six attributes. In contrast to the abalone data, besides the *temperature* of the patient which is measured on a continuous scale, all other attributes are binary taking the values *yes* or *no*: occurrence of *nausea, lumbar pain, urine pushing, micturition pain, burning or swelling*. The data contains two binary attributes which can be used as evaluation attributes, the diagnosis *acute inflammation of the urinary bladder* (in the following called Evaluation Attribute 1) and the diagnosis *acute nephritis* (Evaluation Attribute 2). Both diseases can but must not co-occur.

Scenic detects five clusters on this data set. This clustering has an NMI of 0.24 with respect to Evaluation Attribute 1 and an NMI of 0.43 with respect to Evaluation Attribute 2. In particular, Cluster 1 consists of 29 patients suffering from *acute nephritis*, 19 of which also suffer from *acute inflammation of the urinary bladder*. Characteristic for this cluster is the high average *temperature* of 40.6 degrees Celsius and the presence of *nausea*. Cluster 2 consists of 21 patients with *acute nephritis* who do not have *inflammation of the urinary bladder*. The patients in this cluster also have a high average *temperature* of 39.9 degrees but in contrast to those in Cluster 1 do not suffer from *nausea* and also not from *micturition pain*, which makes sense since they do not have *inflammation of the bladder*. Cluster 3 consists of 20 patients suffering from *inflammation of the urinary bladder* but without *nephritis*. These patients are characterized by a normal average *temperature* of 36.9 degrees, and by the fact that they are all having the characteristic symptoms of *pushing*, *micturition pain* and *burning or swelling*. Cluster 4 consists of 20 subjects who have neither *nephritis* nor *inflammation of the bladder*. Cluster 5 consists of 30 subjects which are mostly suffering from *inflammation of the bladder* (20 of 30). None of the subjects in Cluster 5 has *nephritis*. In contrast to Cluster 2, the average *temperature* in Cluster 5 is with 38,5 degrees slightly elevated. Furthermore, in contrast to the subjects in Cluster 3, this



(a) Cluster 1, Nephritis (b) Cluster 5, no Nephritis

Figure 6: Embedding of Selected Clusters of Acute Inflammations Data. In sub-figures (ii) filled dots represent patients suffering from *inflammation of the bladder*, open dots represent subjects not suffering from this condition.

group is not suffering from *burning or swelling*. Figure 6 displays the embedding of the attributes (i) and of the objects (ii) for Cluster 1 (a) and Cluster 5 (b). As previously, for clarity of presentation, we only show those categories which show a variation inside the corresponding cluster, since all remaining categories are aligned at the origin of the display. Clusters 1 and 5 are the only clusters which are not class-pure with respect to Evaluation Attribute 1, which means that these clusters contain subjects suffering from *inflammation of the bladder* as well as subjects who are not suffering from this condition. In Cluster 1, we observe a strong dependency among the categories *pushing* and *burning* and *no pushing* and *no burning*, respectively. These pairs of categories are grouped together and aligned at the x-axis of the display of Figure 6(a,i). From Figure 6(a,ii) it becomes evident that this axis separates subjects affected from *inflammation of the bladder* from subjects who are not. In the embedding of the objects we can distinguish among three sub-groups. Sub-group 1 has no symptoms and relatively low *temperature* and does not suffer from i*nflammation of the bladder*. Sub-groups 2 and 3 suffer from this condition, where sub-group 2 has *no burning* but *pushing* and sub-group 3 has both of those characteristic symptoms. The second axis of the display is spanned mainly by the numerical attribute *temperature*, which has slight dependency to the categorical attributes. Especially patients with the symptoms *pushing* and *burning*, i.e. those who have *nephritis* and *inflammation of the bladder* tend to have a higher *temperature* than patients only having *nephritis*. In Cluster 5, we observe different dependencies. Similar as in Cluster 1, the categories *pushing* and *no pushing* follow approximately the x-axis of the display. However, in contrast to Cluster 1, the binary attribute *micturition pain* strongly depends on the *temperature* and vice versa. Subjects without *micturition pain* have higher *temperature* than the other subjects. As in Cluster 1, the x-axis of the display clearly distinguishes subjects with *inflammation of the bladder* from those without. In Figure 6(b,ii) the subjects form three sub-groups which are class-pure with respect to Evaluation Attribute 1. Sub-groups 1 and 2 consists of patients with *inflammation of the bladder*, where the subjects in sub-group 1 have the symptoms *micturition pain* and *pushing*, and the subjects in sub-group 2 only *pushing* but a higher *temperature*. Sub-group 3 is composed of patients who have an elevated *temperature* but no other symptoms. This group has not been diagnosed with *inflammation of the bladder*.

INCONCO detects 4 clusters with an NMI of 0.11 with respect to Evaluation Attribute 1 and 0.52 with respect to Evaluation Attribute 2. As for Scenic, all clusters detected by INCONCO are class-pure with respect to Evaluation Attribute 2. The only difference explaining the lower NMI of Scenic is the fact that Scenic detects one more cluster than INCONCO. Since the evaluation attributes are binary detecting more clusters has negative effects on NMI. Therefore, we parameterized K-means Mixed to detect 2 clusters. However, the result does not match well any of the two evaluation attributes: K-means Mixed has an NMI of 0.007 with respect to Evaluation Attribute 1 and 0.20 with respect to Evaluation Attribute 2.

In summary, Scenic is the only technique yielding a result which matches well both evaluation attributes. The categorical cluster attribute produced by Scenic well corresponds to the diagnosis *nephritis*. Furthermore, in the leading dimensions of the cluster-specific object embedding, the objects are perfectly separated according to the diagnosis *inflammation of the bladder*.

5. RELATED WORK AND DISCUSSION

In clustering numerical data, considering attribute dependencies has a long history with a lot of well-known approaches like ORCLUS [1] or CURLER [14]. The research area and is often referred to as correlation clustering or generalized subspace clustering. Compared to the large volume of research papers on clustering numerical data, only relatively few approaches focus on categorical data. Recently, some approaches for finding clusters in subspaces of categorical data sets have been proposed like Clicks [17].

However, to the best of our knowledge, the topic of generalized subspace clustering on categorical data is largely unexplored.

Despite the practical relevance integrative mining of data with different measurement scales, only disproportionably few approaches focus on clustering data represented by numerical and categorical attributes. Recently, some approaches to clustering such mixed-type data sets have been proposed, in particular K-prototypes [9], CFIKP [16], CAVE [8], CEBMDC [7], INTEGRATE [3], K-means Mixed [2] and INCONCO [12]. In early approaches like K-prototypes, not only the number of clusters k but also the relative importance of the numerical and categorical attributes in clustering needs to be specified by input parameters. Many later papers set the primary focus on the key question how to balance the relative importance of numerical and categorical information in clustering and developed creative solutions: Ensemble methods in CEBMDC, complex optimization methods in K-means Mixed and elements of information theory in INTEGRATE, CAVE and INCONCO.

In the experimental evaluation, we decided to compare Scenic to K-means Mixed [2] and INCONCO [12], since both algorithms follow different philosophies on how to integrate numerical and categorical information in clustering. K-means Mixed dynamically learns the significance of each single attribute during the clustering process. The objective function of the algorithm formalizes the idea that attributes well separating the data objects are more interesting for clustering. By this attribute weighting scheme, K-means Mixed performs some kind of primitive subspace clustering. To the best of our knowledge, INCONCO is the first step towards making the benefits of generalized subspace clustering available for mixed-type data. As Scenic, INCONCO is based on the MDL Principle however with a different intention. INCONCO does not regard clustering as an unsupervised classification problem but directly compresses the input data. Mixed-type attribute dependencies are revealed by an extended Cholesky Decomposition. However, as mentioned in the experimental section, the cluster model of INCONCO is limited to support certain types of attribute dependencies only. In particular, all categories of all nominal variables involved in a dependency with some numerical variables need to have different Gaussian distributions of the numerical variables. Our experiments on synthetic and real data demonstrate that this condition does often not hold. As additional major benefit over INCONCO, the result of Scenic comprises a clear visualization of the attribute dependencies which is very helpful for interpretation.

6. CONCLUSION

In this paper, we have proposed Scenic, a technique for generalized dependency clustering across measurement scales. The key idea of Scenic is to consider clustering as an unsupervised classification problem. Scenic clusters by constructing low-dimensional joint embeddings of the objects and mixed-type attributes which allow to predict the original attribute values with high accuracy. To avoid overfitting, we combine this idea with data compression. Guided by the Minimum Description Length Principle, Scenic automatically selects the number of clusters as well as the dimensionality of the embeddings. As an additional value-add, the embeddings of objects and attributes are a valuable source of information for interpretation. Especially the embedding of the attributes allows understanding even complex dependencies involving several categorical and numerical attributes at first glance.

Many open challenges remain in mining heterogeneous data: Scenic assigns each object to one distinct cluster. However, objects may belong to several clusters. We therefore focus on fuzzy and subspace clustering with special attention on detecting interesting non-redundant clusterings. Besides clustering, we currently focus on outlier detection methods respecting the dependencies among numerical and categorical attributes. As a long term goal, we want to include further types of attributes, e.g. hierarchical and relational information.

7. ACKNOWLEDGMENTS

C.P. is supported by the Alexander von Humboldt Foundation.

8. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *SIGMOD*, pages 70–81, 2000.

[2] A. Ahmad and L. Dey. A k-mean clustering algorithm for mixed numeric and categorical data. *Data Knowl. Eng.*, 63(2):503–527, 2007.

[3] C. Böhm, S. Goebl, A. Oswald, C. Plant, M. Plavinski, and B. Wackersreuther. Integrative parameter-free clustering of data with mixed type attributes. In *PAKDD*, pages 38–47, 2010.

[4] C. Böhm, K. Kailing, P. Kröger, and A. Zimek. Computing clusters of correlation connected objects. In *SIGMOD*, pages 455–466, 2004.

[5] J. V. Davis and I. Dhillon. Differential entropic clustering of multivariate gaussians. In *NIPS*, 2006.

[6] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[7] Z. He, X. Xu, and S. Deng. Clustering mixed numeric and categorical data: A cluster ensemble approach. *CoRR*, abs/cs/0509011, 2005.

[8] C.-C. Hsu and Y.-C. Chen. Mining of mixed data with application to catalog marketing. *Expert Syst. Appl.*, 32(1):12–23, 2007.

[9] Z. Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.*, 2(3):283–304, 1998.

[10] U. V. Luxburg, M. Belkin, O. Bousquet, and Pertinence. A tutorial on spectral clustering. *Stat. Comput*, 2007.

[11] G. Michailidis and J. de Leeuw. The gifi system of descriptive multivariate analysis. *STATISTICAL SCIENCE*, 13:307–336, 1998.

[12] C. Plant and C. Böhm. Inconco: interpretable clustering of numerical and categorical objects. In *KDD*, pages 1127–1135, 2011.

[13] J. Rissanen. *Information and Complexity in Statistical Modeling*. Springer, 2007.

[14] A. K. Tung, X. Xu, and B. C. Ooi. CURLER: Finding and visualizing nonlinear correlation clusters. In *SIGMOD*, pages 467–478, 2005.

[15] N. X. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *ICML*, pages 1073–1080, 2009.

[16] J. Yin and Z. Tan. Clustering mixed type attributes in large dataset. In *ISPA*, pages 655–661, 2005.

[17] M. J. Zaki, M. Peters, I. Assent, and T. Seidl. Clicks: An effective algorithm for mining subspace clusters in categorical datasets. *Data Knowl. Eng.*, 60(1):51–70, 2007.