Integrative Parameter-Free Clustering of Data with Mixed Type Attributes

Christian Böhm[1], Sebastian Goebl[1], Annahita Oswald[1], Claudia Plant[2], Michael Plavinski[1], and Bianca Wackersreuther[1]

[1] University of Munich
[2] Technische Universität München
{boehm,oswald,wackersreuther}@dbs.ifi.lmu.de,
{goebl,plavinsk}@cip.ifi.lmu.de,
plant@lrz.tum.de

Abstract. Integrative mining of heterogeneous data is one of the major challenges for data mining in the next decade. We address the problem of integrative clustering of data with mixed type attributes. Most existing solutions suffer from one or both of the following drawbacks: Either they require input parameters which are difficult to estimate, or/and they do not adequately support mixed type attributes. Our technique INTEGRATE is a novel clustering approach that truly integrates the information provided by heterogeneous numerical and categorical attributes. Originating from information theory, the Minimum Description Length (MDL) principle allows a unified view on numerical and categorical information and thus naturally balances the influence of both sources of information in clustering. Moreover, supported by the MDL principle, parameter-free clustering can be performed which enhances the usability of INTEGRATE on real world data. Extensive experiments demonstrate the effectiveness of INTEGRATE in exploiting numerical and categorical information for clustering. As an efficient iterative algorithm INTEGRATE is scalable to large data sets.

1 Introduction

Integrative data mining is among the top 10 challenging problems in data mining identified in [1]. Moreover it is essential for solving many of the other top 10 challenges, including data mining in social networks and data mining for biological and environmental problems. In this paper, we focus on *integrative clustering*. Clustering aims at finding a natural partitioning of the data set into meaningful groups or clusters. Thus, clustering provides an overview on major patterns in the data without requiring much previous knowledge. During the last decades, clustering has attracted a lot of attention as reflected in a huge volume of research papers, e.g. [2,3,4,5,6,7], to mention a few. We address the question of how to find a natural clustering of data with mixed type attributes. In everyday life, huge amounts of such data are collected, for example from credit assessments. The collected data include numerical attributes (e.g. credit amount, age), as well as categorical attributes (e.g. personal status). A cluster analysis of credit assessment data is interesting, e.g., for target marketing. However, finding a natural clustering of such data is a non-trivial task. We identified two major problems: Either much previous knowledge is required, or there is no adequate support of mixed type attributes.

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 38–47, 2010.

To cope with these two major problems, we propose INTEGRATE, a parameter-free technique for integrative clustering of data with mixed type attributes. The major benefits of our approach, which to the best of our knowledge no other clustering method meets all of them, can be summarized as follows:

- Natural balance of numerical and categorical information in clustering supported by information theory;
- Parameter-free clustering;
- Making most effective usage of numerical as well as categorical information;
- Scalability to large data sets.

The rest of this paper is organized as follows: Section 2 gives a brief survey of the large previous work. Section 3 presents a detailed derivation of $iMDL$, an information-theoretic clustering quality criterion suitable for integrative clustering. Section 4 presents our effective and efficient iterative algorithm INTEGRATE optimizing $iMDL$. Section 5 documents that INTEGRATE makes most effective usage of numerical as well as categorical information by comparing it to well-known and state-of-the-art clustering algorithms on synthetic and real data sets. Section 6 summarizes the paper.

2 Related Work

The algorithm k-prototypes [3] combines k-means [2] for clustering numerical data with k-modes for categorical data in order to cluster mixed type data. The attribute weights and the number of clusters have to be determined a priori. CFIKP [8] can process large data sets by k-prototypes in combination with a CF*-tree, which pre-clusters the data into dense regions. The problem for selecting the number of clusters remains. The algorithm CAVE [9] is an incremental entropy-based method which first selects k clusters, parametrized by the user, and then assignes objects to these clusters based on variance and entropy. Knowledge of the similarity among categorical attributes is needed in order to construct the distance hierarchy for the categorical attributes. The cluster ensemble approach CEBMDC [10] overcomes the problem of selecting k but requires a threshold parameter that defines the intra-cluster similarity between objects. The CBC algorithm [11] is an extension of BIRCH [4] for clustering mixed type data. It uses a weight-balanced tree that needs two paramters, defining the number of entries for (non)-leaf nodes. Furthermore, all entries in a leaf node must satisfy a particular threshold requirement. Ahmad and Dey [12] propose a k-means-based method for mixed type attributes, but the process of solving the optimization of the cost function is very complex and thus not scalable to large data sets. [13] uses standard fuzzy c-means on a set of features which is mapped to a set of feature vectors with only real valued components. This mapping is computationally intensive and is designed rather for low dimensional data. An extension of the cost function of entropy weighting k-means [14] to more efficiently specify the inter- and intra-cluster similarities is proposed by the IWEKM approach [15]. Some papers have focused on avoiding the choice of k in partitioning clustering, e.g. X-Means [5], RIC [6] and OCI [7]. However, these clustering methods are designed for numerical vector data only.

3 Minimum Description Length for Integrative Clustering

Notations. 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. For a categorical attribute A, we denote a possible value of A by a. The result of our algorithm is a disjoint partitioning of DS into k clusters $\mathcal{C}_1, ..., \mathcal{C}_k$.

Likelihood and Data Compression. One of the most challenging problems in clustering data with mixed attribute type is selecting a suitable distance function, or unifying clustering results obtained on the different representations of the data. Often, the weighting between the different attribute types needs to be specified by parameter settings, cf. Section 2. The minimum description length (MDL) principle provides an theoretical foundation for parameter-free integrative clustering avoiding this problem. Regarding clustering as a data compression problem allows us a unifying view, naturally balancing the influence of categorical and numerical attributes in clustering. Probably the most important idea of MDL which allows integrative clustering is relating the concepts of likelihood and data compression. Data compression can be maximized by assigning short descriptions to regular data objects which exhibit the characteristic patterns and longer descriptions to the few irregular objects or outliers.

3.1 Coding Categorical Data

Assume a data set, where each object is represented by one categorical attribute A with two possible values. It can be shown that the code length to encode this data is lower bounded by the entropy of A. Thus, the coding costs CC of A are provided by:

$$CC(A) = -\sum_{a \in A} p(a) \cdot \log_2 p(a).$$

By the application of the binary logarithm we obtain the code length in bits. If we have no additional knowledge on the data we have to assume that the probabilities for each value are equal. Hence, we need one bit per data object. Clustering, however, provides high-level knowledge on the data which allows for a much more effective way to reduce the costs. Even if the probabilities for the different outcomes of the attributes are approximately equal w.r.t. the whole data set, often different clusters with non-uniform probabilities can be found. As an example, refer to Figure 1. The data are represented by two numerical attributes (which we ignore for the moment) and one categorical attribute which has two possible values, red and $blue$. Considering all objects, the probabilities for red and $blue$ are equal. However, it is evident that the outcomes are not uniformly distributed. Rather, we have two clusters, one preliminarily hosts the red objects, and the other the blue ones. In fact, the data has been generated such that in the left cluster, we have 88% of blue objects and 12% of red objects. For the right cluster, the ratio has been selected reciprocally. This clustering drastically reduces the entropy and hence $CC(A)$ = 0.53 bits per data object, which corresponds to the entropy of A in both clusters.

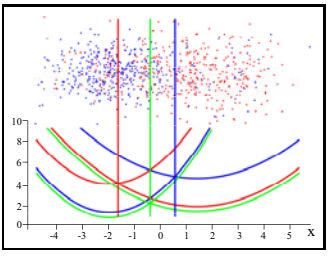


Fig. 1. Top: Example data set with two numerical and one categorical attribute with the outcomes *red* and *blue*. Bottom: Cost curves assuming two clusters: Considering the numerical information only (green), integrating numerical and categorical information (red, blue). For each outcome and each cluster, we have a unique cost curve. Intersection points mark the resulting cluster borders.

3.2 Coding Numerical Data

To specify the probability of each data object considering an additional numerical attribute B, we assign a probability density function (PDF) to B. In this paper, we apply a Gaussian PDF for each numerical attribute. However, let us note that our ideas can be straightforwardly extended to other types PDF, e.g. Laplacian or Generalized Gaussian. Thus, the PDF of a numerical attribute B is provided by:

$$p(b) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(b-\mu_B)^2}{2\sigma_B^2}\right).$$

If the data distribution of B is Gaussian with mean μ_B and standard deviation σ_B, we minimize the costs of the data by a coding scheme which assigns short bit strings to objects with coordinate values that are in the area of high probability and longer bit strings to objects with lower probability. This principle is also commonly referred to as Huffman-Coding. The coding costs CC of B are provided by:

$$CC(B) = -\int p(b)\log_2\ p(b)\mathbf{d}b.$$

Again, if we have no knowledge on the data, we would have to assume that each attribute is represented by a single Gaussian with mean and standard deviation determined from all data objects. As discussed for categorical data, clustering can often drastically reduce the costs. Most importantly, relating clustering to data compression allows us a unified view on data with mixed type attributes. Consider again the data displayed in Figure 1. In addition to the categorical attribute we now consider the numerical x-coordinate, denoted by X. To facilitate presentation, we ignore the y-coordinate which is processed analogously. The two green curves at the bottom represent the coding costs of the two clusters considering X. For both curves, the cost minimum coincides with the mean of the Gaussian which generated the data. The cluster on the right has been generated with slightly larger variance, resulting in slightly higher coding costs. The intersection of both cost curves represents the border between the two clusters provided by X, indicated by a green vertical line. In addition, for each cluster and each outcome

of the categorical attribute, we have included a cost curve (displayed in the corresponding colors). Again, the intersection points mark the cluster borders provided by the categorical attribute. Consider, e.g., the red vertical line. Red objects with a value in X beyond that point are assigned to the cluster on the right. Thus, in the area between the red and the blue vertical line, the categorical value is the key information for clustering. Note that all borders are not fixed but optimized during the run of our algorithm.

3.3 A Coding Scheme for Integrative Clustering

We also need to elaborate a coding scheme describing the clustering result itself. The additional costs for encoding the clustering result can be classified into two categories: the parameter costs PC required to specify the cluster model and the id-costs IDC required to specify the cluster-id for each object, i.e. the information to which cluster the object belongs.

For the parameter costs, lets focus on the set of objects belonging to a single cluster $\mathcal{C}$. To specify the cluster model, we need for each categorical attribute A to encode the probability of each value or outcome a. For a categorical attribute with $|A|$ possible values, we need to encode $|A|-1$ probabilities since the remaining probability is implicitly specified. For each numerical attribute B we need to encode the parameters μ_B and σ_B of the PDF. Following a central result form the theory of MDL [16], the parameter costs to model the $|\mathcal{C}|$ objects of the cluster can be approximated by $p/2 \cdot \log_2 |\mathcal{C}|$, where p denotes the number of parameters. The parameter costs depend logarithmically on the number of objects in the cluster. The considerations behind this are that for clusters with few objects, the parameters do not need to be coded with very high precision. To summarize, the parameter costs for a cluster $\mathcal{C}$ are provided by

$$PC(\mathcal{C}) = \frac{1}{2} \cdot ((\sum_{A_{cat}} |A| - 1) + |B_{num}| \cdot 2)) \cdot \log_2 |\mathcal{C}|.$$

Here A_{cat} stands for all categorical attributes and B_{num} for all numerical attributes in the data. Besides the parameter costs, we need for each object to encode the information to which of the k clusters it belongs. Also for the id-costs, we apply the principle of Huffman coding which implies that we assign shorter bitstrings to the larger clusters. Thus, the id-costs of a cluster $\mathcal{C}$ are provided by:

$$IDC(\mathcal{C}) = \log_2 \frac{n}{|\mathcal{C}|}.$$

Putting it all together, we are now ready to define $iMDL$, our information-theoretic optimization goal for integrative clustering.

$$iMDL = \sum_{\mathcal{C}} (\sum_{A} |\mathcal{C}| \cdot CC(A)) + PC(\mathcal{C}) + IDC(\mathcal{C}).$$

For all clusters $\mathcal{C}$ we sum up the coding costs for all numerical and categorical attributes A. To these costs we need to add the parameter costs and the id-cost of the cluster, denoted by $PC(\mathcal{C})$ and $IDC(\mathcal{C})$, respectively. Finally, we sum up these three terms for all clusters.

4 The Algorithm INTEGRATE

Now we present the highly effective algorithm INTEGRATE for clustering mixed type attributes that is based on our new MDL criterion $iMDL$, defined in Section 3. INTEGRATE is designed to find the optimal clustering of a data set DS, where each object x comprises both numerical and categorical attributes by optimizing the overall compression rate. First, INTEGRATE builds an initial partitioning of k clusters. Each cluster is represented by a Gaussian PDF in each numerical dimension B with μ_B and σ_B, and a probability for each value of the categorical attributes. All objects are then assigned to the k clusters by minimizing the overall coding costs $iMDL$. In the next step, the parameters of each cluster are recalculated according to the assigned objects. That implies the μ and the σ in each numerical dimension and the probabilities for each value of the categorical attributes, respectively. After initialization the following steps are performed repeatedly until convergence. First, the costs for coding the actual cluster partition are determined. Second, assignment of objects to clusters is performed in order to decrease the $iMDL$ value. Third, the new parameters of each cluster are recalculated. INTEGRATE terminates if no further changes of cluster assignments occur. Finally, we receive the optimal clustering for DS represented by k clusters according to minimum coding costs.

4.1 Initialization

The effectiveness of an algorithm often heavily depends on the quality of the initialization, as it is often the case that the algorithm can get stuck in a local optimum. Hence, we propose an initialization scheme to avoid this effect. We have to find initial cluster representatives that correspond best to the final representatives. An established method for partitioning methods is to initialize with randomly chosen objects of DS. We adopt this idea and take the μ of the numerical attributes of k randomly chosen objects as cluster representatives. During initialization, we set $\sigma = 1.0$ in each numerical dimension. The probabilities of the values for the categorical attributes are set to $\frac{1}{|a|}$. Then a random set of $\frac{1}{z}n$ objects is selected, where n is the size of DS and z = 10 turned out to give satisfying results. Finally, we chose the clustering result that minimizes $iMDL$ best, within m initialization runs. Typically $m = 100$ runs suffice for an effective result. As only a fraction of DS is used for the initialization procedure, our method is not only effective but also very efficient.

4.2 Automatically Selecting the Number of Clusters k

Now we propose a further improvement of the effectiveness of INTEGRATE. Using $iMDL$ for mixed type data we can avoid the parameter k. As an optimal clustering that represents the underlying data structure best has minimum coding costs, $iMDL$ can also be used to detect the number of clusters. For this purpose, INTEGRATE uses $iMDL$ no longer exclusively as selection criterion for finding the correct object to cluster assignment. Rather we now estimate the coding costs for each k where k is selected in a range of $1 \leq k \leq n$. For efficiency reasons INTEGRATE performs this iteration step on a $z\%$ sample of DS. The global minimum of this cost function gives the optimal k and thus the optimal number of clusters.

5 Experimental Evaluation

Since INTEGRATE is a hybrid approach combining the benefits of clustering methods using only numeric attributes and those for categorical attributes we compare algorithms of both categories and algorithms that can also handle mixed type attributes. In particular, we selected the popular k-means algorithm, the widely used method k-modes, the k-means-based method by Ahmad and Dey denoted by KMM and k-prototypes (cf. Section 2). For k-means and k-modes the numerical and categorical attributes were ignored. For evaluation we used the validity measure by [17] referred to as DOM in the following (smaller values indicate a better cluster quality), which has the advantage that it allows for clusterings with different numbers of clusters and integrates the class labels as "ground truth". We report in each experiment the average performance of all clustering algorithms over 10 runs.

5.1 Synthetic Data

If not otherwise specified the artificial data sets include three Gaussian clusters with each object having two numerical attributes and one categorical attribute. To validate the results we added a class label to each object which was not used for clustering.

Varying Ratio of Categorical Attribute Values. In this experiment we generated three-dimensional synthetic data sets with 1,500 points including two numerical and one two-valued categorical attribute. We varied the ratio for each of the values of the categorical attributes from 1:0 to 0:1 clusterwise in each data set. Without need for difficult parameter setting INTEGRATE performs best in all cases (cf. Figure 2). Even in the case of equally (5:5) distributed values, where the categorical attribute gives no information for separating the objects, the cluster quality of INTEGRATE is best compared to all other methods. As k-means does not take the categorical attributes into account the performance is relatively constant.

Varying Variance of Clusters. This experiment aims at comparing the performance of the different methods on data sets with varying variances. In particular, we generated synthetic data sets each comprising 1,500 points including two numerical and one two-valued categorical attribute that form three Gaussian clusters with a variance ranging from 0.5 to 2.0. Figure 3 shows that INTEGRATE outperforms all competitors in all cases, in which each case reflects different degree of overlap of the three clusters. Even at a variance of 2.0 where the numerical attributes carry nearly no cluster information our proposed method shows best cluster quality as in this case the categorical attributes are used to separate the clusters. On the contrary, k-modes performs worst as it can only use the categorical attribute as single source for clustering.

Varying Clustersize. In order to test the performance of the different methods on data sets with unbalanced clustersize we generated three Gaussian clusters with different variance and varied the ratio of number of points per cluster from 1:10:1 to 10:1:10 in five steps. It is obvious from Figure 4 that INETGRATE separates the three clusters best even with highly unbalanced cluster sizes. Only in the case of two very small clusters and one big cluster (1:10:1) k-modes shows a slightly better cluster validity.

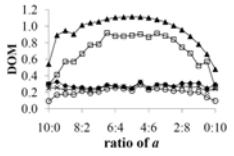


Fig. 2. Varying ratio of categorical attribute values

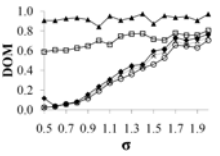


Fig. 3. Varying variance of clusters

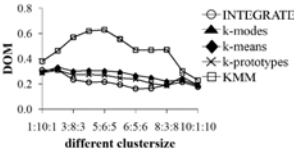


Fig. 4. Varying Clustersize

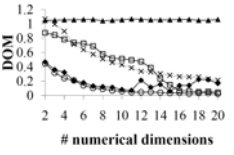


Fig. 5. Numerical dimensions

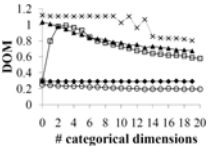


Fig. 6. Categorical dimensions

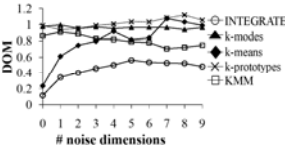


Fig. 7. Noise dimensions

Varying Number of Numerical Dimensions. In this experiment we leave the number of categorical attributes to a constant value and successively add numerical dimensions to each object that are generated with a variance of σ=1.8. INTEGRATE shows best performance in all cases (cf. Figure 5). All methods show a slight increase in cluster quality when varying the numerical dimensionality, except k-modes that performs constantly as it does not consider the numerical attributes.

Varying Number of Categorical Dimensions. For each object we added three-valued categorical attributes where we set the probablity of the first value to 0.6 and the probability of the two remaining values to 0.2, respectively. Figure 6 illustrates that our proposed method outperforms the other methods and even k-modes by magnitudes which is a well-known method for clustering categorical data. Whereas KMM shows a heavy decrease in clustering quality in the case of two and four additional categorical attributes, our method performs relatively constant. Taking the numerical attributes not into account the cluster validity of k-means remains constant.

Noise Dimensions. Figure 7 illustrates the performance of the different methods on noisy data. It it obvious that INTEGRATE outperfoms all compared methods when adding non-clustered noise dimensions to the data. k-means shows a highly increase in the DOM values which refers to decreasing cluster validity. Even in the case of nine noise dimensions INTEGRATE leads to the best clustering result.

5.2 Real Data

Finally, we show the practical application of INTEGRATE on real world data, available at the UCI repository `http://archive.ics.uci.edu/ml/`. We chose two different data sets with mixed numerical and categorical attributes. An additional class attribute allows for an evaluation of the results. Table 1 reports the μ and σ of all methods within 10 runs. For all compared methods we set k to the number of classes.

Table 1. Results on real data

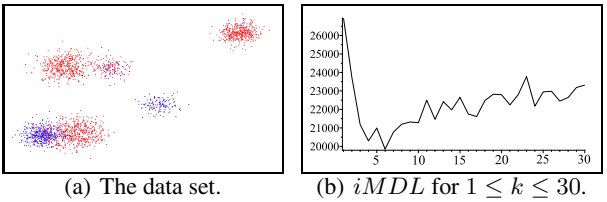
		INTEGRATE	**k-means**	**k-modes**	**kMM**	**k-prototypes**
Heart Disease	μ	1.23	1.33	1.26	1.24	1.33
	σ	0.02	0.01	0.03	0.02	0.00
Credit Approval	μ	0.61	0.66	0.70	0.63	0.66
	σ	0.03	0.00	0.00	0.09	0.00

Heart Disease. The Heart-Disease data set comprises 303 instances with six numerical and eight categorical attributes each labeled to an integer value between 0 and 4 which refers to the presence of heart disease. Without any prior knowledge on the data set we obtained best cluster validity of 1.23 with INTEGRATE. KMM performed slighty worse. However, the runtime of INTEGRATE is 0.1 seconds compared to KMM which took 2.8 seconds to return the result.

Credit Approval. The Credit Approval data set contains results of credit card applications. It has 690 instances, each being described by six numerical and nine categorical attributes and classified to the two classes 'yes' or 'no'. With a mean DOM value of 0.61 INTEGRATE separated the objects best into two clusters in only 0.1 seconds without any need for setting input parameters.

5.3 Finding the Optimal k

On the basis of the data set illustrated in Figure 8(a) we highlight the benefit of INTEGRATE for finding the correct number of clusters that are present in the data set. The data set comprises six Gaussian clusters with each object having two numerical attributes and one categorical attribute with two different values that are marked in "red" and "blue", respectively. Figure 8(b) shows the $iMDL$ of the data model for different values of k. The cost function has its global minimum, which refers to the optimal number of clusters, at k = 6. In the range of $1 \leq k \leq 4$ the plotted function shows an intense decrease in the coding costs and for $k > 6$ a slight increase of the coding costs as in these cases the data does not optimally fit into the data model and therefore causes high coding costs. Note, that there is a local minimum at k = 4 which would also refer to a meaningful number of clusters.



(a) The data set. (b) $iMDL$ for $1 \leq k \leq 30$.

Fig. 8. Coding costs for different k according to a data set that consists of $k = 6$ clusters

6 Conclusion

We have introduced a new information-theoretic clustering method — INTEGRATE. We gave a solution to avoid difficult parameter settings guided by the information-theoretic idea of data compression. We have shown with extensive experiments that INTEGRATE uses the numerical and categorical information most effectively. And finally, INTEGRATE shows high efficiency and is therefore scalable to large data sets.

References

1. Yang, Q., Wu, X.: 10 challenging problems in data mining research. IJITDM 5(4), 597–604 (2006)
2. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
3. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Min. Knowl. Discov. 2(3), 283–304 (1998)
4. Zhang, T., Ramakrishnan, R., Livny, M.: Birch: An efficient data clustering method for very large databases. In: SIGMOD Conference, pp. 103–114 (1996)
5. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML, pp. 727–734 (2000)
6. Böhm, C., Faloutsos, C., Pan, J.Y., Plant, C.: Robust information-theoretic clustering. In: KDD, pp. 65–75 (2006)
7. Böhm, C., Faloutsos, C., Plant, C.: Outlier-robust clustering using independent components. In: SIGMOD Conference, pp. 185–198 (2008)
8. Yin, J., Tan, Z.: Clustering mixed type attributes in large dataset. In: ISPA, pp. 655–661 (2005)
9. Hsu, C.C., Chen, Y.C.: Mining of mixed data with application to catalog marketing. Expert Syst. Appl. 32(1), 12–23 (2007)
10. He, Z., Xu, X., Deng, S.: Clustering mixed numeric and categorical data: A cluster ensemble approach. CoRR abs/cs/0509011 (2005)
11. Rendon, E., Sánchez, J.S.: Clustering based on compressed data for categorical and mixed attributes. In: SSPR/SPR, pp. 817–825 (2006)
12. Ahmad, A., Dey, L.: A k-mean clustering algorithm for mixed numeric and categorical data. Data Knowl. Eng. 63(2), 503–527 (2007)
13. Brouwer, R.K.: Clustering feature vectors with mixed numerical and categorical attributes. IJCIS 1-4, 285–298 (2008)
14. Jing, L., Ng, M.K., Huang, J.Z.: An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. IEEE Trans. Knowl. Data Eng. 19(8), 1026–1041 (2007)
15. Li, T., Chen, Y.: A weight entropy k-means algorithm for clustering dataset with mixed numeric and categorical data. In: FSKD 2008, vol. (1), pp. 36–41 (2008)
16. Rissanen, J.: An introduction to the mdl principle. Technical report, Helsinkin Institute for Information Technology (2005)
17. Dom, B.: An information-theoretic external cluster-validity measure. In: UAI, pp. 137–145 (2002)