Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference

MDL-Based Unsupervised Attribute Ranking

Zdravko Markov
Central Connecticut State University, 1615 Stanley Street, New Britain, CT 06050, USA
markovz@ccsu.edu

Abstract

In the present paper we propose an unsupervised attribute ranking method based on evaluating the quality of clustering that each attribute produces by partitioning the data into subsets according to its values. We use the Minimum Description Length (MDL) principle to evaluate the quality of clustering and describe an algorithm for attribute ranking and a related clustering algorithm. Both algorithms are empirically evaluated on benchmark data sets. The experiments show that the MDL-based ranking performs closely to the supervised information gain ranking and thus improves the performance of the EM and k-means clustering algorithms in purely unsupervised setting.

Introduction

Attribute (feature) selection helps learning algorithms to perform better with noisy or irrelevant attributes, and improves their efficiency with imbalanced and sparse data. Attribute *subset selection* is computationally expensive and intractable in the worst case when it requires the exhaustive search of 2^m attribute sets for m attributes. Therefore an approach often used in practice, called *attribute weighting*, is to assume that attributes are independent (although this rarely happens with real data) and evaluate not attribute sets, but individual attributes, rank them by relevance to the learning task and then pick a number of attributes from the top of the ranked list.

Attributes may be selected both in the presence (*supervised* setting) and absence (*unsupervised* setting) of class labels. Attribute selection is more popular in supervised learning and classification because labeled data allow the use of good evaluation measures for the quality of the selected attributes, which are usually based on the accuracy of predicting class labels. Thus the goal of supervised attribute selection is to *find the smallest set of attributes that will maximize predictive accuracy*.

Unsupervised attribute selection is a difficult problem not only because it is hard to solve in the absence of prior knowledge (class labels), but also because it is generally not well defined. The definition that we are using here is based on the one provided in (Dy and Brodley 2004):

> The goal of feature selection for unsupervised learning is to find the smallest feature subset that best uncovers interesting natural groupings (clusters) from data according to the chosen criterion.

As we take the weighting approach, we consider the problem of *evaluating each attribute with respect to its ability to uncover interesting natural groupings in data*. We measure the interestingness of these groupings by applying a clustering quality criterion to the clusters that each attribute produces by partitioning the data into subsets according to its values, similarly to the divide-and-conquer technique used in decision tree learning. For measuring the quality of clustering we use the Minimum Description Length (MDL) principle originally suggested by Rissanen (1978). In this paper we describe an efficient algorithm for MDL-based attribute ranking and a related clustering algorithm, which are empirically evaluated on benchmark data.

Related Work

There are two basic strategies for supervised attribute selection – *wrapper* and *filter* methods. The wrapper methods evaluate attributes by running learning algorithms to create prediction models and use the predictive accuracy of these models to measure the attribute relevance to the prediction task. The filter approaches directly measure the ability of the attributes to determine the class labels using statistical correlation, information metrics, probabilistic or other methods.

There are numerous approaches to supervised feature selection. We refer readers to (Liu and Motoda 2008) for a comprehensive coverage of the algorithms in this area. This book also includes a chapter on unsupervised feature

selection. A classification of methods for feature selection and a short survey of recent work in the area is provided in (Liu et al. 2010). It covers both supervised and unsupervised approaches. In general, however, the supervised methods are much more popular than the unsupervised ones. This is also the case with their practical implementations and availability in software systems. For example, the popular Weka machine learning suit (Hall et al. 2009) provides 17 attribute evaluation algorithms, which may be combined with 11 search methods, while no algorithms at all are available under the unsupervised category. Nevertheless, unsupervised attribute selection is an active research area, mainly focusing on the attribute evaluation techniques in the framework of clustering. The main reason for this is that clustering approaches provide evaluation measures that can be used in the wrapper framework (replacing predictive accuracy used in the supervised setting).

The basic idea of the wrapper approaches to unsupervised attribute selection is to evaluate a subset of attributes by the quality of clustering obtained by using these attributes. Devaney and Ram (1997) use the category utility function and the COBWEB conceptual clustering algorithm to evaluate subsets of attributes. Dy and Brodley (2004) explore the EM framework and use the scatter separability and maximum likelihood evaluation functions to find feature subsets. In (Mitra et al. 2002) an algorithm for feature selection is proposed that uses partitioning the data into clusters such that the features within the clusters are highly similar. A feature is selected from each cluster thus forming the reduced feature subset.

Filter methods are also studied in unsupervised learning. Many of those methods explore classical statistical methods for dimensionality reduction, like Principal Component Analysis (PCA) and maximum variance. (Note that PCA transforms the features instead of selecting a subset of the original ones.). Although the maximum variance criteria finds features that are useful for representing data, they may not be as useful for discriminating between data in different clusters. This idea is explored in (Deng 2010) to create an algorithm that selects the features that best preserve the multi-cluster structure of the data.

Another group of methods use information-based measures. A method called entropy reduction is proposed by Dash and Liu (2000). It measures the entropy in data based on a normalized distance between pairs of instances and evaluates attributes by the reduction of the entropy when the attribute is removed from data. This and other methods are empirically evaluated for the purposes of text clustering in (Liu et al. 2003). Most of the unsupervised methods discussed in this study use specific evaluation techniques related to the bag-of-words representation of text documents. This study also suggests that attribute selection can improve the quality and efficiency of text clustering.

There is a third, *embedded* approach that lies between the wrapper and filter approaches. In (Guan et al. 2011) feature selection and clustering are incorporated in one algorithm. Their method is based on a hierarchical beta-Bernoulli prior combined with a Dirichlet process mixture and allows learning both the number of clusters and the number of features to retain.

Our approach falls in the framework of wrapper approaches. It is similar in spirit to the one described in (Mitra et al. 2002), which uses an information compression measure to create clusters with highly similar features and selects a feature from each cluster. However we partition the data into clusters using the natural splitting created by the attribute values, similarly to the divide-and-conquer technique used in decision tree learning. As we use MDL, our approach may be traced back to the classical work by Quinlan and Rivest (1989). They use MDL to select the attributes that partition the dataset during the process of creating a decision tree, however in the presence of class labels. There are clustering approaches that use MDL for clustering model evaluation. These approaches differ in the specific MDL encoding scheme and in the way clusters are generated. In (Kontkanen et al. 2005) MDL is used to evaluate grouping of data items that can be compressed well together, so that the total code length over all data groups is optimized. Thus an efficient compression indicates underlying regularities that are common to all members of a group, which in turn may be used as an implicitly defined similarity metric for the purposes of clustering. In (Lai et al. 2009) a distance based clustering technique using MDL for evaluating clusters is proposed. Our approach also uses MDL for clustering model evaluation. It is based on a simple and efficient encoding scheme and creates clusters by using the divide-and-conquer technique of decision tree learning.

MDL-based clustering model evaluation

Clustering may be seen as searching for patterns or finding regularity in data. We consider each possible clustering as a *hypothesis H* that *describes* (*explains*) the data *D* in terms of frequent patterns or regularities. Then we apply the MDL principle to evaluate the possible hypotheses. Formally, we have to compute the *description length* of the data $L(D)$, the hypothesis $L(H)$, and the data given the hypothesis $L(D|H)$. The interpretation of $L(H)$ and $L(D)$ is the minimum number of bits needed to encode (or communicate) the hypothesis and the data respectively, while $L(D|H)$ represents the number of bits needed to encode D if we already know H. The latter term makes a great deal of sense if we think of H as a pattern that repeats

in D. Once we know that pattern we don't have to encode all its occurrences, rather we encode only the pattern itself and the differences that identify each individual instance in D. Thus the more regularity in data the shorter description length $L(D|H)$. In addition, we have to balance this term with the description length of H itself, because it will greatly depend on the complexity of its pattern. For example, if H describes the data *exactly* (H includes a pattern for each data instance) then $L(D|H)$ will be 0, but $L(H)$ will be large, in fact equal to the description length of the data itself $L(D)$. In terms of clustering this means a set of singleton clusters equivalent to the original data set. We can also put all data instances in one cluster. Then H will be empty and $L(H)=0$, but $L(D|H)$ will include the code length of every single data instance thus making $L(D|H)=L(D)$. This suggests that the best hypothesis should minimize the sum $L(H)+L(D|H)$ (MDL principle), or alternatively maximize $L(D)-L(H)-L(D|H)$ (*information compression* principle).

The key to applying MDL is to find an encoding scheme for the hypotheses and the data given the hypotheses. The encoding should be the same for both, because we are adding code lengths and want to keep the balance between them. One simple scheme used for this purpose is based on the assumption that hypotheses and data are uniformly distributed and the probability of occurrence of an item out of n alternatives is $1/n$. Thus the minimum code length of the message informing us that a particular item has occurred is equal to $-log_2 1/n = log_2 n$. To compute this, given a particular *description language* we need to count all possible hypotheses and data instances given each hypothesis. Hereafter we use the *attribute-value* description language with *nominal attributes*, which allows us to compute easily discrete probabilities and code length by using attribute-value pair counts.

Let us consider a data set D and a set T containing all (different) attribute-value pairs in D, i.e. $T = \bigcup_{X \in D} X$, where X is a data instance (a set of attribute-values). We define the description language as *all subsets of T*. Let us also consider a clustering of the data into n clusters $\{C_1,C_2,\ldots,C_n\}$. The hypothesis H producing this clustering is defined as a set of rules R_i, each one covering (explaining) the instances in cluster C_i and assigning to each of them cluster label i. R_i can be represented by the set T_i of all (different) attribute-value pairs that occur in the instances from cluster C_i:

R_i: If $X \subseteq T_i$ Then $X \in C_i$

Assume that there are k different attribute-value pairs in the description language ($k = |T|$) and k_i different attribute value pairs in each cluster C_i ($k_i = |T_i|$). Then, representing the left-hand side of the rule is equivalent to selecting k_i attribute-value pairs out of k possible. The number of choices for this selection is equal to the number of k_i-combinations of k elements. The right-hand side of the rule represents the selection of one out of n cluster labels. Thus the code length of each rule is

$$L(R_i) = \log_2 \binom{k}{k_i} + \log_2 n$$

and the code length of the hypothesis H is

$$L(H) = \sum_{i=1}^{n} L(R_i)$$

The description length of the data given the hypothesis $L(D|H)$ is also a sum of the corresponding MDL terms applied to the rules:

$$L(D \mid H) = \sum_{i=1}^{n} L(C_i \mid R_i)$$

To define $L(C_i|R_i)$ we need to estimate the probability of occurrence of each instance X in C_i. Let $|X| = m$ (the number of attributes in our language). Knowing R_i means that we know the set of attribute-value pairs T_i representing the rule (i.e. they have already been communicated). Then the occurrence of each X in C_i is equivalent of selecting m attribute-value pairs out of k_i ($k_i = |T_i|$). Thus the message about the occurrence of all data instances in C_i has the following description length:

$$L(C_i \mid R_i) = |C_i| \times \log_2 \binom{k_i}{m}$$

The minimum description length of hypothesis H is

$$MDL(H) = L(H) + L(D \mid H) = \sum_{i=1}^{n} L(k, k_i, m, n), \qquad (1)$$

where

$$L(k, k_i, m, n) = \log_2 \binom{k}{k_i} + \log_2 n + |C_i| \times \log_2 \binom{k_i}{m} \qquad (2)$$

The function $L(k,k_i,m,n)$ computes the MDL of rule R_i, but the parameters k, k_i, m, n are derived directly from the given clustering $\{C_1,C_2,\ldots,C_n\}$. In fact, we introduced the hypothesis H and the rules R_i only to justify our encoding scheme. So, we may assume that this function actually computes the MDL of cluster C_i and the sum in (1) computes the MDL of the given clustering.

$$MDL(\{C_1, C_2, \ldots, C_n\}) = \sum_{i=1}^{n} L(k, k_i, m, n) \qquad (3)$$

Illustrative Example

Let us consider the classical "play tennis" data set (Quinlan 1986) shown in Table 1. For the purposes of clustering we ignore the class (attribute play) and the ID attribute (used only to identify instances), which leaves four attributes (m=4) with 3, 3, 2, and 2 values each. Thus we have a total of 10 attribute-value pairs (k=10).

ID	outlook	temp	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Table 1. *"Play tennis" data set*

Let us compute $MDL(\{C_1,C_2\})$, where C_1 = {1, 2, 3, 4, 8, 12, 14} and C_2 = {5, 6, 7, 9, 10, 11, 13}. These clusters are obtained by splitting the data by using the values of the humidity attribute – high in C_1 and normal in C_2. The sets of attribute-value pairs occurring in each cluster are: T_1 = {outlook=sunny, outlook=overcast, outlook=rainy, temp=hot, temp=mild, humidity=high, windy=false, windy=true}, T_2 = {outlook=sunny, outlook=overcast, outlook=rainy, temp=hot, temp=mild, temp=cool, humidity=normal, windy=false, windy=true}. Thus $k_1 = |T_1| = 8$ and $k_2 = |T_2| = 9$, $k = 10$, $m = 4$, and $n = 2$. Plugging these values in formula (2) gives:

$$MDL\ (C_1) = \log_2 \binom{10}{8} + \log_2 2 + 7 \times \log_2 \binom{8}{4} = 49.39$$

$$MDL\ (C_2) = \log_2 \binom{10}{9} + \log_2 2 + 7 \times \log_2 \binom{9}{4} = 53.16$$

Thus $MDL(\{C_1,C_2\})$=102.55 bits. Also, $MDL(C_1) < MDL(C_2)$, which shows that C_1 exhibits more regularity than C_2 as it has fewer number of attribute-value pairs. Similarly we evaluate the remaining attributes and rank them accordingly: temp – 101.87, humidity – 102.56, outlook – 103.46, and windy – 106.33. This ranking reflects the quality of clustering that each attribute produces and is a good measure of *attribute relevance* for clustering and classification.

Efficient Implementation

Algorithm 1 shows our implementation of MDL attribute ranking. The key to the efficiency of this algorithm is in the computation of k_{ij}, the number of attribute values in cluster c_{ij}. We compute it by using a set V_{ij}^{l} for each value j of each attribute A_i, collecting the values of all attributes A_l ($l = 1,..., m$) so that the sum of the cardinalities of V_{ij}^{l} over l produces the value of k_{ij}. Updating these sets is the basic computational step in the algorithm, which occurs m^2 times for each instance. When implemented by a hash table, this step takes a constant time. Thus the overall time complexity of the algorithm is $O(nm^2)$, where n is the number of instances, and m is the number of attributes. Its space complexity is $O(pm^2)$, where p is the maximal number of values an attribute can take.

---

1. Denote:
   m = number of attributes.
   x_{ij} = the j^{th} value of attribute A_i.
   $C_{ij} = \{X \mid x_{ij} \in X\}$, where X is a data instance.
   n_i = the number of values of attribute A_i.
2. Let $V_{ij}^{l} = \varnothing$, $l = 1,..., m$ (initialization).
3. For each instance $X = \langle x_{1j_1}, x_{2j_2},..., x_{mj_m} \rangle$
   For each $l = 1,..., m$
   For each $i = 1,..., m$
   $V_{ij}^{l} = V_{ij}^{l} \cup \{x_{ij}\}$.
4. For each $i = 1,..., m$ (compute MDL)
   $k_{ij} = \sum_{l=1}^{m} |V_{ij}^{l}|$
   $k = \sum_{i=1}^{m} n_i$
   $MDL\ (A_i) = \sum_{i=1}^{n_i} L(k, k_{ij}, m, n_i)$
5. Order attributes by $MDL\ (A_i)$

Algorithm 1. *MDL-Ranker*

The algorithm has two basic advantages – it is *linear* in the number of instances and *incremental* in terms of processing instances. Each data instance is processed only once (in step 3) and there is no need to store it after that. This eliminates the need of storing the entire data set in the memory and allows processing of very large data sets. Our Java implementation is able to process the trec data set (see Table 2) with 13195 attributes and 3204 instances in 3 minutes on a 3GHz Intel-based PC.

MDL-based Clustering

Our hierarchical clustering algorithm (Algorithm 2) splits the data using the values of the top MDL ranked attribute and then applies recursively the same technique to the resulting clusters. At each split based on attribute A it computes the sum of the compression *Comp(A) = L(D)-L(H)-L(D|H)* in the resulting subclusters. Starting from the root this sum initially increases at each split indicating an increase in the quality of the hypotheses describing the subclusters. If at some level of the tree the sum starts to decrease (indicating that no better clustering can be found) the algorithm forms a leaf of the clustering tree.

Function *MDL-Cluster(D)*

1. Choose attribute $A = \arg\min_i MDL(A_i)$
2. Let A take values $x_1, x_2, ..., x_n$
3. Split data $D = \bigcup_{i=1}^{n} C_i$, where $C_i = \{X \mid x_i \in X\}$
4. If $Comp(A) > \sum_{i=1}^{n} Comp(C_i)$ then stop. Return *D*.
5. For each $i = 1,..., n$ Call *MDL-Cluster*(C_i)

Algorithm 2. *MDL-Cluster*

Experimental Evaluation

We evaluate our algorithms on the data sets summarized in Table 2. The first four are text classification data, which are very sparse, have a larger number of attributes than instances, and also varying number of class labels. The reuters-2class and reuters-3class data sets are extracted from reuters by using the instances of the two and three largest classes. The rest of the data sets are popular benchmark data, which we downloaded from the Weka project website (Hall et al. 2009). We also use the Weka Data Mining system for discretization of the numeric attributes of the iris and ionosphere data sets, for information gain attribute ranking, and for the EM and k-means clustering experiments. Java implementations of our algorithms along with all data sets are available at http://www.cs.ccsu.edu/~markov/DMWsoftware.zip.

Data Set	Instances	Attributes	Classes
reuters	1504	2887	13
reuters-3class	1146	2887	3
reuters-2class	927	2887	2
trec	3204	13195	6
soybean	683	36	19
soybean-small	47	36	4
iris	150	5	3
ionosphere	351	35	2

Table 2. *Data sets used for evaluation*

Attribute Ranking

In this experiment we compare the MDL ranking with one supervised (InfoGain) and two unsupervised ranking approaches. The first unsupervised approach is similar to the MDL approach – it evaluates the quality of clustering produced by splitting the data by the attribute values using the *sum of squared errors*. The second approach, called *entropy reduction*, is described in (Dash and Liu 2000). It evaluates each attribute by the reduction of the entropy in data when the attribute is removed. To compare the rankings we use the *average precision* measure known from information retrieval, which is defined as follows:

$$\text{Average Precision} = \frac{1}{|D_q|}\sum_{k=1}^{|D|} r_k \times \text{PrecisionAtRank(k)}$$

$$\text{PrecisionAtRank(k)} = \frac{1}{k}\sum_{i=1}^{k} r_i \qquad r_i = \begin{cases} 1 & \text{if } a_i \in D_q \\ 0 & \text{otherwise} \end{cases}$$

where D is the set of all attributes and D_q – the set of relevant attributes. The set D_q contains the attributes selected by the Weka's Wrapper Subset Evaluator using the Naïve Bayes classifier with Linear-Forward-Selection search. The index of r_i represents the rank of the attribute in the ranking produced by the algorithm that we evaluate. The results from this experiment are shown in Table 3. The MDL ranking outperforms the two unsupervised ranking algorithms and in some cases gets closer to the supervised InfoGain method. The latter is an additional advantage of MDL as it does not use the class information, which is essential for InfoGain. Thus the closeness to the InfoGain performance is an indication of the consistency of class labeling with the natural groupings of instances based only on attribute-value patterns (it's known that soybean, iris and ionosphere have well separated classes).

Data set	$\lvert D_q \rvert$	InfoGain	MDL	Error	Entropy
reuters	15	0.3183	0.1435	0.0642	0.0030
reuters-3class	10	0.3948	0.1852	0.1257	0.0027
reuters-2class	7	0.5016	0.2438	0.1788	0.3073
trec	14	0.4890	0.2144	0.0637	0.0010
soybean	16	0.6265	0.5606	0.3871	0.4152
soybean-small	2	0.6428	0.3500	0.0913	0.1213
iris	1	1.0000	1.0000	1.0000	0.3333
ionosphere	9	0.6596	0.5041	0.2575	0.4252

Table 3. *Average precision of attribute ranking*

Clustering

Table 4 shows the results from clustering the data sets by using the MDL-Cluster, EM and k-means algorithms. As all data sets are labeled we evaluate the accuracy of clustering by the *classes-to-clusters* measure used in the Weka system (Hall et al. 2009). It's a comparison to the "true" cluster membership specified by the class attribute and is computed as the total number of majority labels in the leaves divided by the total number of instances. The Weka implementations of EM and k-means are used with the default settings for all parameters except for the number of clusters (set to the known number of classes) and the random number generator seed (three different seeds are used and the accuracy results are averaged).

Data set	EM		k-Means		MDL-Cluster	
	Acc. (%)	No. of Clusters	Acc. (%)	No. of Clusters	Acc. (%)	No. of Clusters
reuters	43	6	31	13	59	12
reuters-3class	58	3	48	3	73	7
reuters-2class	71	2	61	2	90	7
trec	26	6	29	6	44	11
soybean	60	19	51	19	51	7
soybean-small	100	4	91	4	83	4
iris	95	3	69	3	96	3
ionosphere	89	2	81	2	80	3

Table 4. *Classes to clusters evaluation accuracies*

Attribute Selection

In this experiment we rank the attributes by using the MDL-Ranker and InfoGain algorithms and then run EM and k-means with decreasing number of attributes selected from the top of the ranked lists. By measuring the classes-to-clusters accuracy we compare the performance of each one of the two clustering algorithms on the data sets with MDL ranked attributes and with InfoGain ranked attributes. Some of the results are shown in Figures 1–4. Due to lack of space we include only the results from the large text data sets and one graph with EM results as they are similar to those from k-means.

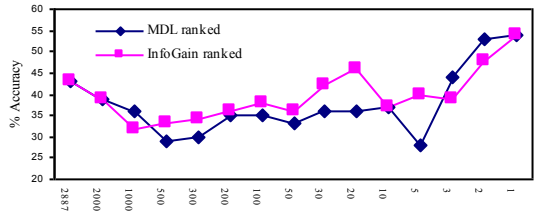

Figure 1. *EM accuracy with reuters data*

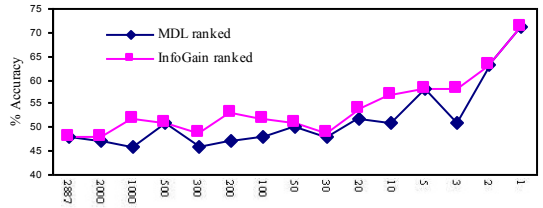

Figure 2. *k-means accuracy with reuters data*

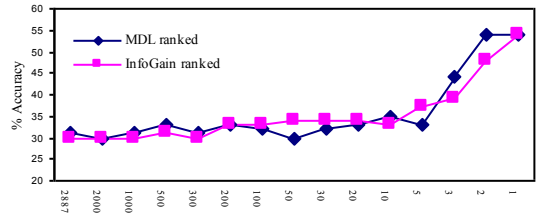

Figure 3. *k-means accuracy with reuters-3class data*

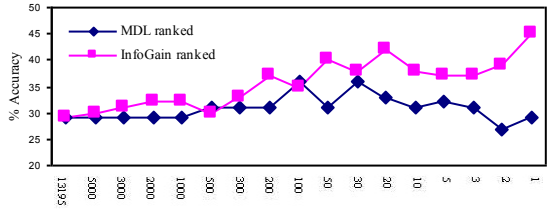

Figure 4. *k-means accuracy with trec data*

Conclusion

In this paper, we introduced an MDL-based measure that evaluates clustering quality and presented algorithms that use this measure for unsupervised attribute ranking and clustering. We evaluated the MDL clustering algorithm on benchmark data and showed that it outperforms the EM and k-means algorithms on most of them. The experiments with attribute selection showed that the MDL-based ranking without class information performs closely to the InfoGain method, which essentially uses class information. Thus, our approach can improve the performance of clustering algorithms in purely unsupervised setting.

References

Dash, M. and Liu, H. Feature selection for clustering. Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000, pp. 110–121.

Deng, C., Chiyuan, Z., Xiaofei, H. Unsupervised Feature Selection for Multi-Cluster Data, KDD'10, July 25–28, 2010, Washington, DC, USA.

Devaney, M. and Ram, A. Efficient Feature Selection in Conceptual Clustering, Proceedings of the 14th International Conference on Machine Learning (ICML 1997), pp. 92-97, Nashville, 1997, Morgan Kaufmann.

Dy, J. G. and Brodley, C. E. Feature Selection for Unsupervised Learning. Journal of Machine Learning Research 5: 845-889, 2004.

Guan, Y., Dy, J. G., and Jordan, M. A Unifed Probabilistic Model for Global and Local Unsupervised Feature Selection, Proceedings of the 28-th International Conference on Machine Learning, Bellevue, WA, 2011.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The WEKA Data Mining Software: An Update, SIGKDD Explorations, Volume 11, Issue 1, 2009.

Lai, P.-H., O'Sullivan, J. A. and Pless, R. Proceedings of the 2009 IEEE International Symposium on Information Theory (ISIT), IEEE Press, Piscataway, NJ, USA pp. 1318-1322, 2009.

Liu, T., Liu, S., Chen, Z., and Ma, W. An Evaluation on Feature Selection for Text Clustering. Proceedings of the 12th International Conference on Machine Learning (ICML-2003), 488-495, Washington DC, 2003.

Huan Liu, H., Motoda, H., Setiono, R., and Zheng, Z. Feature Selection: An Ever Evolving Frontier in Data Mining, JMLR: Workshop and Conference Proceedings 10: 4-13, The Fourth Workshop on Feature Selection in Data Mining, Hyderabad, India, 2010.

Kontkanen, P., Myllymaki, P., Buntine, W., Rissanen, J, and Tirri, H. An MDL framework for Data Clustering, in Grünwald, P., Myung, J. and Pitt, M. (Eds.), Advances in Minimum Description Length: Theory and Applications, MIT Press, 2005.

Liu, H. and Motoda, H. Computational Methods of Feature Selection, Chapman & Hall/CRC, 2008.

Mitra, P., C. Murthy, C., and Pal S. Unsupervised feature selection using feature similarity, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(3):301-312, 2002.

Quinlan, J. R. Induction of Decision Trees. Machine Learning 1(1): 81-106, 1986.

Quinlan, J. R. and Rivest, R. Inferring Decision Trees Using the Minimum Description Length Principle, Information and Computation, Vol. 80, 1989, 227 - 248.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.