A New MDL-based Clustering Algorithm

Zdravko Markov
Computer Science Department
Central Connecticut State University New Britain, CT
markovz@ccsu.edu

Abstract

The Minimum Description Length (MDL) principle is commonly used to evaluate machine learning models. In clustering, it may be used to assess the overall quality of data clustering or in the implementation of clustering algorithms for guiding the search for possible clusterings. In this paper, we explore the latter approach. We describe a new clustering algorithm, which uses an MDL-based measure during the process of creating clusters. The algorithm processes the data instances to be clustered one at a time and uses them to form new clusters or assigns them to already created ones. Instead of using distances (as in other popular algorithms) this process is controlled by evaluating the overall clustering quality at each step based on the MDL measure. The basic advantages of the algorithm are its simplicity, lack of adjustable parameters, and low computational complexity. The computation of the MDL score for the current clustering, which is performed for each instance, doesn't depend on the number of instances in the clusters and is linear with the number of attributes. The performance of the algorithm is evaluated on benchmark datasets.

Introduction

To assess clustering algorithms various criteria are used. Criterion functions are also used in the implementation of clustering algorithms for guiding the search of possible clusterings. In this paper we explore this approach. We first define a clustering evaluation measure based on the Minimum Description Length (MDL) principle. Given a particular clustering of the data, it provides a numeric score representing the number of bits to encode the clustering model. Thus, according to Occam's Razon concept, the lower the number of bits is, the better the clustering quality is. Then, we propose an algorithm that works similarly to the Canopy clustering algorithm (McCallum et al. 2000), where instances to be clustered are removed from the data set one at a time and used to create new clusters or assigned to already created ones. Instead of using distances (as in the Canopy algorithm), in our algorithm this process is controlled by evaluating the overall clustering quality at each step based on the defined MDL measure. We illustrate the performance of the algorithm on a few datasets from the UCI ML repository compared to two traditional clustering algorithms, k-means and EM.

Related Work

All approaches to clustering use some kind of evaluation measure for the clustering quality and differ in the way clusters are created so that the clustering quality is maximized. In (Markov 2015) the authors present a hierarchical clustering algorithm based on using selected attributes to split data in clusters. To choose the attributes they use an unsupervised attribute ranking scheme based on the Minimum Description Length (MDL) principle described in (Markov 2013). There are other approaches that also integrate clustering with attribute selection. In (Devaney and Ram 1997) the category utility function and the Cobweb clustering algorithm are used for this purpose. In (Dy and Brodley 2004) the authors explore the EM framework and use the scatter separability and maximum likelihood evaluation functions to find proper subsets of attributes. Another group of methods uses information-based measures. A method called entropy reduction is proposed in (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). The approach of (Markov 2013) 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. The use of MDL may be traced back to the classical work (Quinlan and Rivest 1989), which uses MDL to select the attributes that partition the dataset during the

process of creating a decision tree, however in the presence of class labels.

Other approaches use MDL for clustering model evaluation. These approaches differ in the specific MDL encoding scheme, and in the way, clusters are generated. In (Kontkanen 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, efficient compression indicates underlying regularities that are common to all members of a group, which in turn may be used as a 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; however, it creates clusters in an incremental manner by processing data instances one at a time and using them to create new clusters or assigning them to already created ones. In this respect it is similar to the Canopy clustering algorithm (McCallum et al. 2000), however instead of using distances, in our algorithm this process is controlled by evaluating the overall clustering quality at each step based on an MDL measure.

MDL-based Clustering Evaluation

Consider an information theory setting for encoding (or communicating) a dataset *D* with the shortest message using the attribute-value description language. Assume that there are *m* attributes and *k* different attribute-value pairs that occur in *D*, i.e. $m = |x|$ and $k = |\cup_{x \epsilon D}\, x|$, where *x* is a data instance. Encoding *D* means encoding each instance $x \epsilon D$, which in turn means choosing a subset of *m* attribute-value pairs out of *k* possible ones. As the number of choices is equal to the number of *m*-combinations of a set with *k* elements, the probability of this choice is

$$p(x) = \frac{1}{\binom{k}{m}}$$

(assuming that data are uniformly distributed). According to information theory the minimal code length (in bits) of the message about the occurrence of instance *x* in data is

$$L(x) = -\log_2 p\,(x) = \log_2 \binom{k}{m}$$

To get the minimal code length of the data we multiply the minimal code length of *x* by the number of instances, i.e.

$$L(D) = |D| \times \log_2 \binom{k}{m}$$

Let us consider a clustering of the data into *n* clusters, i.e. $D = C_1 \cup C_2 \cup \ldots \cup C_n$. This clustering may be viewed as an alternative description of the data, a hypothesis *H*. According to the MDL principle (Quinlan and Rivest 1989) the best hypothesis should minimize the code length of the data described by the hypothesis. Formally, this means minimizing the sum of the code length of the hypothesis itself and the code length of the data given the hypothesis, i.e. $L(H)+L(D|H)$, or maximizing the difference $L(D)–L(H)–L(D|H)$, called Information Compression.

Assuming that there are k_i attribute-value pairs in each cluster C_i, we can represent the minimal code length of *H* as

$$L(H) = \sum_{i=1}^{n} \left(\log_2 \binom{k}{k_i} + \log_2 n\right)$$

The term in the sum represents the number of bits needed to encode the description language for each cluster (choosing k_i out of *k* attribute-value pairs) and the cluster label (choosing one out of *n* cluster labels). Once we communicated the hypothesis, encoding each cluster comes to encoding each instance in C_i. Using the same reasoning as for computing $L(D)$ and summing up over all clusters we represent the code length of the data given the hypothesis as

$$L(D|H) = \sum_{i=1}^{n} |C_i| \log_2 \binom{k_i}{m}$$

Thus MDL of *H* (clustering $C_1 \cup C_2 \cup \ldots \cup C_n$) is

$$MDL(C_1 \cup C_2 \cup \ldots \cup C_n) = \sum_{i=1}^{n} MDL(C_i)\ ,$$

where $\sum_{i=1}^{n} MDL(C_i) =$

$$\sum_{i=1}^{n} \left(\log_2 \binom{k}{k_i} + \log_2 n + |C_i| \log_2 \binom{k_i}{m}\right) \qquad (1)$$

Let us discuss an example of computing MDL using the classical play tennis dataset (Mitchell 1997), shown in Table 1 (the play column shows the original class label).

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Table 1. Paly Tennis dataset

In this dataset we have 4 attributes (m=4), and the total number of different attribute-value pairs is k=10 (3 values of outlook, 3 values of temperature, 2 values of humidity, and 2 values of windy). Thus the code length of the data is

$$L(D) = |D| \times \log_2 \binom{k}{m} = 14 \times \log_2 \binom{10}{4} = 108.00$$

Let us consider three clustering models: H_1 – a single cluster containing all instances, H_2 – 14 singleton clusters, each one containing a data instance, and H_3 – the clustering based on the values of attribute temperature. When n=1 formula (1) is reduced to $L(D)$, thus $MDL(H_1) = 108.00$. In

clustering H_2 each instance is a cluster, so k_i = 4 (the number of attributes) and $MDL(H_2)$ is

$$MDL(H_2) = 14 \times \left(\log_2 \binom{10}{4} + \log_2 14 + 1 \times \log_2 \binom{4}{4} \right) = 161.30$$

Hypothesis H_3 has three clusters – C_{hot}, C_{mild}, and C_{cool}, each one containing the instances with the same value of the temperature attribute. The corresponding cluster parameters are k_{hot} = 7, k_{mild} = 8, k_{cool} = 7, $|C_{hot}|$ = 4, $|C_{mild}|$ = 6, $|C_{cool}|$ = 4. The MDL of hypothesis H_3 is computed as:

$$MDL(C_{hot}) = \log_2 \binom{10}{7} + \log_2 3 + 4 \times \log_2 \binom{7}{4} = 29.01$$

$$MDL(C_{mild}) = \log_2 \binom{10}{8} + \log_2 3 + 6 \times \log_2 \binom{8}{4} = 43.85$$

$$MDL(C_{cool}) = \log_2 \binom{10}{7} + \log_2 3 + 4 \times \log_2 \binom{7}{4} = 29.01$$

$$MDL(H_3) = MDL(C_{hot}) + MDL(C_{mild}) + MDL(C_{cool}) = 101.87$$

According to their MDL the best clustering is H_3. Also, it is the only one that provides data compression, i.e. $MDL(H_3) < L(D)$. For the other two the compression is negative or zero, i.e. it takes more information to encode the data using the hypothesis than to encode the data alone.

This example also illustrates the idea of using MDL to create clusters based on the attribute values, which is explored in (Markov 2015). In this paper, we use the same MDL computation, however we create clusters using a different approach. Our algorithm is described in the next section.

MDL Clustering Algorithm

The algorithm is shown in Figure 1. It starts with the initial dataset D, considered as a single cluster.

1. Input data D
2. $C = \emptyset$
3. While $|D| > 0|$ for each $x \epsilon D$ do the following:
   $MDL_1 = MDL(D \backslash x \cup C \cup \{x\})$
   $MDL_2 = MDL(D \backslash x \cup C + x)$
   $D = D \backslash x$
   If $MDL_1 < MDL_2$ then $C = C \cup \{x\}$
   else $C = C + x$
4. Return clustering C

Notes:

- MDL is computed according to Formula (1).
- $C + x$ means inserting x in one of the clusters in C, which minimizes MDL.

Figure 1. MDL clustering algorithm

Then at each step, an instance is removed from D and either added to one of the existing clusters or used to create a new cluster. Thus, for each instance two MDL scores are computed – one, when the instance is removed from D and added to an existing cluster and another one, when the instance is removed from D and used to create a new cluster. Computing these scores requires updating the attribute value counts for each cluster (k_i in Formula 1), which can be done by one pass through the attributes to add or remove their corresponding values. Thus, with m attributes, d data instances, and n clusters, the running time of the algorithm is bounded by $O(mnd)$. This computational complexity is lower than other popular algorithms (like k-means and EM) because our algorithm does not use iterations and convergence.

An example run of the MDL algorithm on the soybean-small dataset (described in the next section) is illustrated in Figure 2.

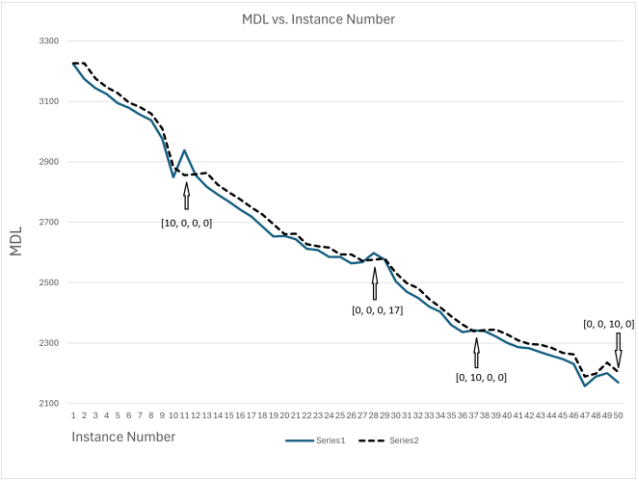

Figure 2. Algorithm performance on soybean-small dataset

The dashed line shows the values of MDL_1 and the solid line – the values of MDL_2. The points when $MDL_1 < MDL_2$ are indicated with arrows and show when a new cluster is added. This cluster is shown by the distribution of the four class labels.

Empirical Evaluation

We evaluated our clustering algorithm on 18 datasets from the UCI ML repository (Bache and Lichman 2013), summarized in Table 2. We selected pure categorical, pure numeric and mixed datasets with various number of attributes and number of instances (deep and shallow data) and also included a very sparse dataset (reuters-2class). In the datasets, indicated with the suffix "–bin", the numeric attributes were discretized into 2-bins. The reuters-2class dataset was created from the original reuters dataset by including the instances from the two largest classes. The class labels were ignored during clustering, and used only for computing the accuracy. Our MDL algorithm was

Dataset	Attributes		Instances	Classes
	Nom	Num		
anneal	32	6	898	5
anneal-bin	32	0	898	5
australian	9	6	690	2
australian-bin	15	0	690	2
breast	0	9	699	2
ecoli	0	7	336	8
german	13	7	1000	2
german-bin	60	0	1000	2
iris	0	4	150	3
iris-3bin	4	0	150	3
lymph	15	3	148	4
lymph-bin	18	0	148	4
reuters-2class	2806	0	853	2
segment	0	19	2310	7
segment-bin	19	0	2310	7
soybean-small	35	0	47	4
soybean	35	0	683	19
vehicle	0	18	846	4

Table 2. Datasets used for evaluation

Dataset	Classification Accuracy			Davies-Bouldin Index			Silhouette Index		
	k-means	EM	MDL	k-means	EM	MDL	k-means	EM	MDL
anneal	0.49	0.55	**0.76**	1.73	1.92	3.52	0.19	0.12	0.06
anneal-bin	0.57	0.51	**0.76**	2.15	2.17	**2.16**	0.24	0.20	0.09
australian	0.79	0.74	0.61	3.71	3.00	4.01	0.14	0.16	0.02
australian-bin	0.73	0.73	**0.73**	1.75	1.75	**1.75**	0.33	0.33	**0.33**
breast	0.96	0.94	0.71	1.07	1.07	3.11	0.40	0.37	0.01
ecoli	0.59	0.67	0.54	2.02	1.86	**1.77**	0.00	0.00	**0.01**
german	0.57	0.63	**0.70**	1.83	2.48	4.85	0.07	0.06	0.01
german-bin	0.56	0.66	**0.70**	4.21	4.22	6.52	0.09	0.02	0.04
iris	0.89	0.91	0.65	2.00	2.00	2.02	0.08	0.08	0.07
iris-3bin	0.94	0.94	0.89	0.48	0.49	0.90	0.66	0.66	0.49
lymph	0.49	0.64	**0.67**	1.43	1.49	2.15	0.14	0.11	0.07
lymph-bin	0.45	0.55	**0.73**	1.49	1.37	2.02	0.11	0.18	0.10
reuters-2class	0.67	0.72	0.65	20.23	31.75	34.44	0.30	0.23	0.05
segment	0.67	0.63	0.58	2.25	2.02	**1.43**	0.00	0.00	**0.04**
segment-bin	0.66	0.60	**0.65**	0.78	0.71	2.08	0.49	0.52	0.26
soybean-small	1.00	1.00	**1.00**	0.79	0.79	**0.79**	0.48	0.48	**0.48**
soybean	0.41	0.39	**0.66**	2.21	2.04	2.25	0.00	0.01	**0.05**
vehicle	0.37	0.37	**0.46**	2.10	1.96	**1.83**	0.03	0.03	**0.04**

Table 3. Evaluation results

compared with the *k*-means and Expectation Maximization (EM) algorithms. We didn't consider our early MDL-based algorithm (Markov 2015) because it was a hierarchical approach, while k-means an EM are partitioning algorithms as ours. We used the Weka system (Hall et al. 2009) and ran the algorithms with their default parameters and the number of clusters set to the known number of classes (ground truth). To minimize their dependence on initialization (the initial clustering) we ran each one 10 times with different random seeds and the clustering, which minimized the sum of squared errors for k-means and maximized the log-likelihood for EM, was used for comparison.

We used three well-known clustering evaluation measures – classification accuracy, Davies-Bouldin Index, and Silhouette Index. The classification accuracy uses the known class labels (ground truth) and is computed as the proportion of instances from the majority class in the clusters (the result from classes-to-clusters evaluation in the Weka system). The Davies-Bouldin Index is a measure of cluster separation and calculates the average similarity between each cluster and its most similar cluster, with lower values indicating better clustering. The Silhouette Index measures how similar an instance is to its own cluster compared to other clusters, ranging from -1 to +1, with higher values indicating better clustering. The results from computing these measures on each datasets are shown in Table 3.

The experimental results show that the performance of our MDL algorithm is generally comparable with *k*-means and EM. It shows good classification accuracy and outperforms k-means and EM on most of the datasets, while its Davies-Boulding and Silhouette scores are relatively lower. This may be explained with the mechanism it uses to create clusters, minimizing their MDL score. This means maximizing the number of shared attribute values in the clusters, which seems consistent with the class labeling of data instances (ground truth). The k-means and EM algorithms try to create clusters with better statistical properties, which are directly measured with the Davies-Boulding and Silhouette metrics. Another trend in the results is that our algorithm performs better on datasets with more nominal than numeric attributes. This is because the MDL measure is more sensitive to data that have more shared attribute values, while the numeric attributes usually have many distinct values. The results from the reuters-2class dataset show that our algorithm can also deal with highly dimensional and sparse data, but still not as well as k-means and EM. This can be explained with the way the cluster MDL score is computed. With high dimensional sparse data, adding or removing a single instance from the cluster changes its score insignificantly. These features of our MDL algorithm need to be further investigated with more extensive experiments and will be addressed in the future studies.

Conclusion

In this paper, we described an MDL-based measure for evaluating clustering quality and presented an algorithm that uses this measure for clustering. We evaluated our algorithm on a set of benchmark data and showed that its

performance is comparable with two traditional clustering algorithms, k-means and EM. The empirical study showed that the algorithm performs well with categorical and low dimensional data and the clustering it creates is close to the ground truth. We also identified the reason why the algorithm doesn't perform well with numeric data, which lies in the mechanism of computing the MDL measure. As originally designed our MDL encoding scheme is based on counting attribute values, which obviously works well with categorical data. In this respect, modifying the MDL encoding scheme to deal with numeric data would be a possible way to improve the algorithm. Our future work will be focused on this direction, along with a deeper analysis of the advantages and drawbacks of our algorithm when applied to bigger and more diverse datasets.

References

Bache, K. and Lichman, M. 2013. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

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

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

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

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

Kontkanen, P., Myllymaki, P., Buntine, W., Rissanen, J, and Tirri, H. 2005. 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.

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

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

Markov, Z. 2013. MDL-based Unsupervised Attribute Ranking, Proceedings of the 26th International Florida Artificial Intelligence Research Society Conference (FLAIRS-26), AAAI Press, 444-449.

Markov, Z. 2015. MDL-Based Hierarchical Clustering, Proceedings of the IEEE 14th International Conference on Machine Learning and Applications (ICMLA-2015), 464-467.

McCallum, A., K. Nigam, K., and Ungar, L.H. 2000. Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching, Proceedings of KDD 2000: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, 169 – 178.

Mitchell, T. 1997. Machine Learning, McGraw Hill.

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

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