A New MDL-based Clustering Algorithm

Zdravko Markov
Computer Science Department
Central Connecticut State University
New Britain, CT 06050, USA
http://www.cs.ccsu.edu/~markov/
markovz@ccsu.edu

A New MDL-based Clustering Algorithm

- Introduction (Clustering)
- Minimum Description Length (MDL)
- MDL for Clustering Model Evaluation
- MDL Clustering Algorithm
- Experimental Evaluation
- Conclusion

Clustering

- Unsupervised learning method: there is no target value (class label) to be predicted, the goal is finding common patterns or grouping similar examples.
- Models / algorithms for clustering:
  - conceptual / partitioning
  - exclusive / overlapping
  - deterministic / probabilistic
  - hierarchical / flat
  - incremental / batch

Clustering Model Evaluation

- External
  - Subjective
  - Using ground truth (labeled data)
- Internal
  - Distance-based
  - Probabilistic
  - MDL-based
- Use
  - Evaluate clustering algorithms
  - Implement clustering algorithms (guide the search for clusters)

Minimum Description Length Principle

Occam's Razor (William of Occam, 14^{th} century)

- *Entities are not to be multiplied beyond necessity*
- *Among several alternatives the simplest one is usually the best choice*.

How do we measure simplicity?

Answer: Minimum Description Length (MDL)

Minimum Description Length Principle

Consider data set *D* and a set of hypotheses $H_1, H_2, \ldots, H_n$, each one describing *D*.

Find the most likely hypothesis $H_i = \arg\max_i P(H_i | D)$

Direct estimation of $P(H_i | D)$ is difficult, so we apply Bayes

$$P(H_i | D) = \frac{P(H_i)P(D | H_i)}{P(D)}$$

Now take –log of both sides

$$-\log_2 P(H_i | D) = -\log_2 P(H_i) - \log_2 P(D | H_i) + \log_2 P(D)$$

Consider hypotheses and data as messages and apply Shannon's information theory, which defines *information in a message as a negative logarithm of its probability*. Then estimate the number of bits *L* needed to encode the messages.

$$L(H_i | D) = L(H_i) + L(D | H_i) - L(D)$$

Minimum Description Length Principle

$L(H_i)$ and $L(D)$ are the minimum number of bits needed to encode the hypothesis and data.

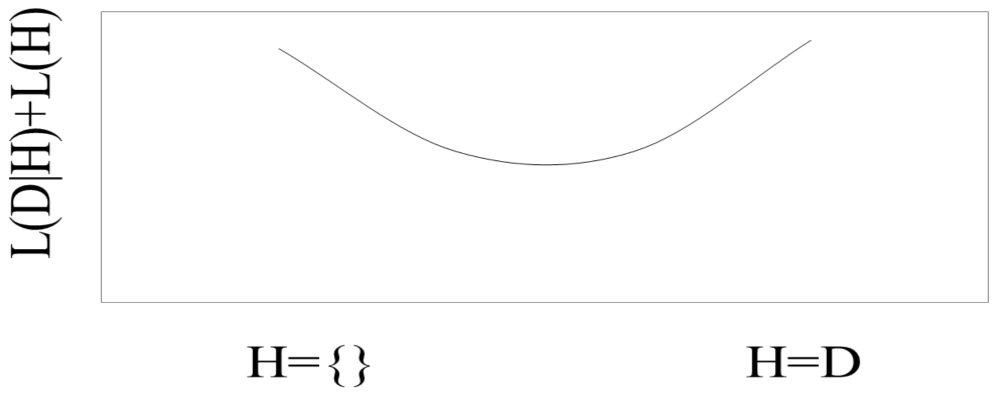
$L(D \mid H_i)$ is the number of bits needed to encode *D* if we know *H*.

If we think of *H* as a pattern that repeats in *D*, 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 *D* the shorter description length $L(D|H)$.

Minimum Description Length Principle

Need a good balance between $L(D|H)$ with $L(H)$, because if H describes the data *exactly* then $L(H) = 0$ but $L(D|H)$ will be large.



We can exclude $L(D)$ as it doesn't depend on the choice of H. Then the best hypothesis H_i should minimize $L(H)+L(D|H)$

$$H_i = \arg\min_i L(H_i) + L(D \mid H_i)$$

Minimum Description Length (MDL) principle
(Quinlan and Rivest 1989)

Encoding MDL for Clustering

Our goal is *encoding a dataset D with the shortest message*.

Assume there are m attributes and k different attribute-value pairs in D, i.e. $m = |x|$ and $k = |\cup_{x \in D} x|$, where x is a data instance.

Encoding each instance $x \in D$ is choosing a subset of m attribute-value pairs out of k possible ones. The probability of this choice (assuming they are uniformly distributed) is

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

Then the code length of x is

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

And the code length of D is

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

Encoding MDL for Clustering

Consider a clustering H that splits the data into n clusters

$$D = C_1 \cup C_2 \cup \ldots \cup C_n.$$

Assuming there are k_i attribute-value pairs in each cluster C_i, the minimal code length of H is

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

The last term in the sum represents encoding of the cluster label (choosing one out of n cluster labels).

Using the same reasoning as for computing $L(D)$ and summing up over all clusters we represent the code length of $L(D|H)$.

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

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

Example: Clustering Play Tennis Data

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

4 attributes (m=4), 10 attribute-value pairs (k=10)

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

Example: Clustering Play Tennis Data

H_1 – single cluster containing all instances. $MDL(H_1) = MDL(D)$ = 108.00.

H_2 – 14 singleton clusters, each one containing a data instance, so $k_i = 4$ (the number of attributes).

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

H_3 – three clusters C_{hot}, C_{mild}, C_{cool}, each one containing the instances with the same value of the temperature attribute. $k_{hot} = 7$, $k_{mild} = 8$, $k_{cool} = 7$, $|C_{hot}| = 4$, $|C_{mild}| = 6$, $|C_{cool}| = 4$.

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

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

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

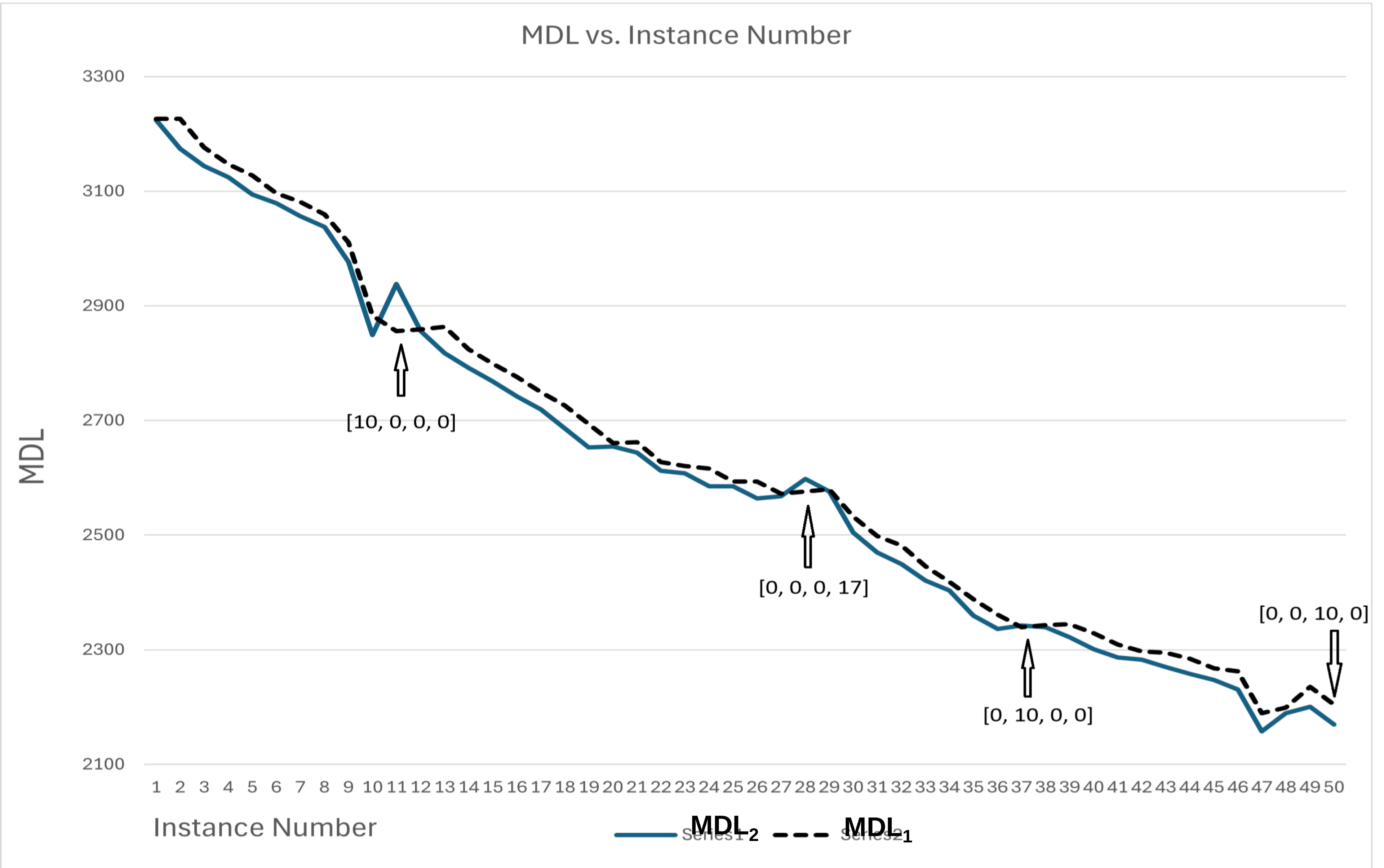
$$MDL(H_3) = MDL(C_{hot}) + MDL(C_{mild}) + MDL(C_{cool}) = 101.87$$

MDL Clustering Algorithm

1. Input data D
2. $\mathrm{C} = \emptyset$
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 $C = C \cup \{x\}$ (add x as a new cluster)
   else $C = C + x$ (insert x in the cluster that minimizes MDL)
4. Return clustering C

Computing MDL_1 and MDL_2 requires updating the attribute value counts (k_i) for each cluster, which is 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 $\mathrm{O}(mnd)$. This computational complexity is lower than other popular algorithms (like k-means and EM) because our algorithm does not use iterations and convergence.

Clustering Soybean-small data (35 attributes, 47 instances, 4 classes)
4 clusters: [10,0,0,0], [0,0,0,17], [0,10,0,0], [0,0,10,0], 100% accuracy



Experimental Evaluation Datasets (UCI ML repository)

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

Experimental Evaluation Metrics

Experiments are run on the Weka system (Hall et al. 2009)

- K-means and EM are run 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.

Evaluation Metrics

- *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).
- *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.
- *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.

Experimental Evaluation Results

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**

Discussion

- 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.
- Scores relatively lower on Davies-Boulding and Silhouette measures. 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.
- 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.

Conclusion

- MDL is simple
- No adjustable parameters
- Low computational complexity
- Works well with categorical data
- Possible improvement: extend MDL measure to numeric data

Thank You!