ECML PKDD 2011 September 5–9, 2011

The Minimum Code Length for Clustering Using the Gray Code

<u>Mahito SUGIYAMA^{†,‡}</u>, Akihiro YAMAMOTO[†] [†]Kyoto University [‡]JSPS Research Fellow

Contributions

- 1. The MCL (Minimum Code Length)
 - A new measure to score clustering results
 - Needed to distinguish each cluster under some fixed encoding scheme for real-valued variables
- 2. **COOL** (<u>CO</u>ding-<u>O</u>riented c<u>L</u>ustering)
 - A general clustering approach
 - Always finds the best clusters (*i.e.*, the global optimal solution) which minimizes the MCL in O(nd)
 - Parameter tuning is not needed
- 3. **G-COOL (COOL with the Gray code)**
 - Achieves internal cohesion and external isolation
 - Finds arbitrary shaped clusters

Demonstration (Synthetic Dataset)



G-COOL



3/26

K-means



3/26

Results (Real datasets)



Results (Real datasets)



Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Clustering Focusing on Compression

- The MDL approach [Kontkanen *et al.*, 2005]
 - Data encoding has to be optimized
 - All encoding schemes are (implicitly) considered
 - The time complexity $\ge O(n^2)$
- The Kolmogorov complexity approach [Cilibrasi, 2005]
 - Measures the distance between data points based on compression of finite sequences
 - Difficult to apply multivariate data
 - Actual clustering process is the traditional agglomerative hierarchical clustering
 - The time complexity $\ge O(n^2)$
- Both approaches are not suitable for massive data

Our Strategy

- *Requirements:*
 - 1. Fast, and linear in the data size
 - 2. Robust to changes in input parameters
 - 3. Can find arbitrary shaped clusters

Our Strategy

- *Requirements:*
 - 1. Fast, and linear in the data size
 - 2. Robust to changes in input parameters
 - 3. Can find arbitrary shaped clusters
- Solutions:
 - 1. Fix an encoding scheme for continuous variables
 - Motivated by *Computable Analysis* [Weihrauch, 2000]
 - 2. Clustering = Discretizing real-valued data
 - Always finds the best results w.r.t. the MCL
 - 3. Use the Gray code for real numbers [Tsuiki, 2002]
 - Discretized data points are overlapped and adjacent clusters are merged

Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

MCL (Minimum Code Length)

- The MCL is the code length of the maximally compressed clusters by using a fixed encoding scheme
- The MCL is calculated in O(nd) by using radix sort
 - *n* and *d* are the number of data and dimension, resp.

MCL (Minimum Code Length)

- The MCL is the code length of the maximally compressed clusters by using a fixed encoding scheme
- The MCL is calculated in O(nd) by using radix sort
 - *n* and *d* are the number of data and dimension, resp.
 - *Example:* $X = \{0.1, 0.2, 0.8, 0.9\},\$

$$\mathcal{C}_1 = \{\{0.1, 0.2\}, \{0.8, 0.9\}\}$$

$$\mathscr{C}_2 = \{\{0.1\}, \{0.2, 0.8\}, \{0.9\}\}$$

- Use binary encoding
- Which is preferred?

Binary Encoding



		A	o ^B o		C D)
id	value					
Α	0.1					
В	0.2					
С	0.8					
D	0.9					
		0	0.25	0.5	0.75	1

		A	o ^B o			\mathbf{i}
id	value					-
Α	0.1					
В	0.2					
С	0.8					
D	0.9					
		0	0.25	0.5	0.75	1



MCL = 1 + 1 = 2



MCL = 1 + 1 = 2

10/26



MCL = 1 + 1 = 2

10/26











Definition of MCL

- Fix an embedding $\gamma : \mathbb{R}^d \to \Sigma^{\omega}$ ($\Sigma = \{0, 1\}$ usually)
- For $p \in range(\gamma)$ and $P \subset range(\gamma)$, define

$$\Phi(p \mid P) := \left\{ w \in \Sigma^* \mid \begin{array}{l} p \in \uparrow w \text{ and } P \cap \uparrow v = \emptyset \text{ for all } v \\ \text{ such that } |v| = |w| \text{ and } p \in \uparrow v \end{array} \right\}$$

- Each element in $\Phi(p | P)$ is a prefix that discriminates p from P

For a partition
$$\mathscr{C} = \{C_1, \dots, C_K\}$$
 of a data set X ,
 $MCL(\mathscr{C}) := \sum_{i \in \{1, \dots, K\}} L_i(\mathscr{C})$, where
 $L_i(\mathscr{C}) := \min \left\{ |W| \middle| \begin{array}{l} \gamma(C_i) \subseteq \uparrow W \text{ and} \\ W \subseteq \bigcup_{x \in C_i} \Phi(\gamma(x) \mid \gamma(X \setminus C_i)) \end{array} \right\}$

Minimizing MCL and Clustering

Clustering under the MCL criterion is to find the global optimal solution that minimizes the MCL

– Find \mathscr{C}_{op} such that

 $\mathscr{C}_{op} \in \operatorname*{argmin}_{\mathscr{C} \in \mathscr{C}(X)_{\geq K}} \mathsf{MCL}(\mathscr{C}),$

where $\mathscr{C}(X)_{\geq K} = \{\mathscr{C} \text{ is a partition of } X \mid \#C \geq K\}$

- We give the lower bound of the number of clusters *K* as a input parameter
 - \mathscr{C}_{op} becomes one set {X} without this assumption

Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Optimization by COOL

- COOL solves the optimization problem in O(nd)
 - *n* and *d* are the number of data and dimension, resp.
 - The naïve approach takes exponential time and space
 - Computing process of the MCL becomes clustering process itself via discretization
- COOL is level-wise, and makes the level-k partition \mathscr{C}^k from k = 1, 2, ..., which holds the following condition:
 - For all $x, y \in X$, they are in the same cluster \iff v = w for some $v \sqsubset \gamma(x)$ and $w \sqsubset \gamma(y)$ with |v| = |w| = k
 - Level-*k* partitions form hierarchy
 - For $C \in \mathscr{C}^k$, there exists $\mathscr{D} \subseteq \mathscr{C}^{k+1}$ such that $\bigcup \mathscr{D} = C$
- For all $C \in \mathscr{C}_{op}$, there exists k such that $C \in \mathscr{C}^{k}$

		Δ	r (
id	value	С С) [0 0		
Α	0.113					
В	0.398					
С	0.526					
D	0.701					
Е	0.796	·				
		0	0.25	0.5	0.75	•

		٨	1	R C		
id	value	\mathcal{C}				
Α	0.113					
В	0.398					
С	0.526					
D	0.701	 				— Lv. 1
Е	0.796		0		1	
		 				I
		0	0.25	0.5	0.75	1





		Δ	F		D	F	
id	value	6		0 0	0	5	
Α	00						
В	01						
С	10	. 00) 0	1	10	11	· LV. Z
D	10	 					→ Lv. 1
Е	11		0			L	
		0	0.25	0.5	0.7	75	1

	<u> </u>	A	B		E	
id	value	(0)	(0)	0 0		
Α	00					
В	01		4	I		
С	10	00	01	10	11	· Lv. 2
D	10	ŀ		ł		→ Lv. 1
Е	11		0		1	
		0 0.	25 0	.5 0.	, 75	1
	$= 2 \cdot 4 = 8$					
COOL with Binary Encoding

		Δ		R C	D	F	
id	value	6`		0 0	0	0	
A	00			ـــــ 1	00 101		Lv. 3
В	01	L		· ـــ	00 101		
С	100	. 0()) 0	1	10	11	· LV. Z
D	101	 					→ Lv. 1
Е	11		0]	L	
		0	0.25	0.5	0.7	75	1

COOL with Binary Encoding

id	value	6	Bo		E C	
A B	00			100 101	1	L۱
C	100	00	01	10	11	→ L\
E	101		0	- - -	1	L\
MCL	= 6 + 6 = 12	0 0.	25 0	.5 0.	75	1

Noise Filtering by COOL

- Noise filtering is easily implemented in COOL
- Define $\mathscr{C}_{\geq N} := \{ C \in \mathscr{C} \mid \#C \geq N \}$ for a partition \mathscr{C}
 - See a cluster C as noises if #C < N
- Example: Given $\mathscr{C} = \{\{0.1\}, \{0.4, 0.5, 0.6\}, \{0.9\}\}$
 - $\mathscr{C}_{\geq 2} = \{\{0.4, 0.5, 0.6\}\}, \text{ and } 0.1 \text{ and } 0.9 \text{ are noises}\}$
- We input the lower bound *N* of the cluster size as a input parameter



Noise Filtering by COOL

- Noise filtering is easily implemented in COOL
- Define $\mathscr{C}_{\geq N} := \{ C \in \mathscr{C} \mid \#C \geq N \}$ for a partition \mathscr{C}
 - See a cluster C as noises if #C < N
- Example: Given $\mathscr{C} = \{\{0.1\}, \{0.4, 0.5, 0.6\}, \{0.9\}\}$
 - $\mathscr{C}_{\geq 2} = \{\{0.4, 0.5, 0.6\}\}, \text{ and } 0.1 \text{ and } 0.9 \text{ are noises}\}$
- We input the lower bound *N* of the cluster size as a input parameter



Algorithm of COOL

Input: A data set X, two lower bounds K and N Output: The optimal partition \mathcal{C}_{op} and noises

function COOL(X, K, N)

- 1: Find partitions $\mathscr{C}^{1}_{\geq N}, \ldots, \mathscr{C}^{m}_{\geq N}$ such that $\|\mathscr{C}^{m-1}_{\geq N}\| < K \leq \|\mathscr{C}^{m}_{\geq N}\|$
- 2: $(\mathscr{C}_{op}, MCL(\mathscr{C}_{op})) \leftarrow FINDCLUSTERS(X, K, \{\mathscr{C}_{\geq N}^{1}, \dots, \mathscr{C}_{\geq N}^{m}\})$
- 3: return ($\mathscr{C}_{op}, X \setminus \bigcup \mathscr{C}_{op}$)
- function FINDCLUSTERS(X, K, $\{\mathscr{C}^1, \ldots, \mathscr{C}^m\}$)
- 1: Find k such that $\|\mathscr{C}^{k-1}\| < K$ and $\|\mathscr{C}^{k}\| \ge K$
- 2: $\mathscr{C}_{op} \leftarrow \mathscr{C}^k$
- 3: if K = 2 then return (\mathscr{C}_{op} , MCL(\mathscr{C}_{op}))
- 4: for each C in $\mathscr{C}^1 \cup \ldots \cup \mathscr{C}^{k-1}$
- 5: $(\mathscr{C}, L) \leftarrow \mathsf{FINDCLUSTERS}(X \setminus C, K 1, \{\mathscr{C}^1, \dots, \mathscr{C}^k\})$
- 6: if $MCL(\mathscr{C} \cup C) < MCL(\mathscr{C}_{op})$ then $\mathscr{C}_{op} \leftarrow C \cup \mathscr{C}$
- 7: return (\mathscr{C}_{op} , MCL(\mathscr{C}_{op}))

Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Gray Code

- Real numbers in [0, 1] are encoded with 0, 1, and \perp Binary: 0.1 \rightarrow 00011 ..., 0.25 \rightarrow 00111 ... Gray: 0.1 \rightarrow 00010 ..., 0.25 \rightarrow 0 \perp 100 ...
- Originally, another binary encoding of natural numbers
 - Especially important in applications of conversion between analog and digital information [Knuth, 2005]
 - The Gray code embedding is an injection γ_G that maps $x \in [0, 1]$ to an infinite sequence $p_0 p_1 p_2 \dots$, where
 - $p_i := 1$ if $2^{-i}m 2^{-(i+1)} < x < 2^{-i}m + 2^{-(i+1)}$ for an odd $m, p_i := 0$ if the same holds for an even m, and $p_i := \bot$ if $x = 2^{-i}m - 2^{-(i+1)}$ for some integer m
 - For a vector $\mathbf{x} = (x^1, ..., x^d), \gamma_G(\mathbf{x}) = p_1^1 ... p_1^d p_2^1 ... p_2^d ...$

Gray Code Embedding



		Δ	1			
id	value	6	, 1	0 0		
Α	0.113					
В	0.398					
С	0.526					
D	0.701					
Е	0.796	·				
		0	0.25	0.5	0.75	-

		Λ	1			
id	value	6	l	0 0		
Α	0.113					
В	0.398					
С	0.526			$\perp 1$		
D	0.701	I	ŀ	+		— Iv. 1
E	0.796		0		1	
		U	0.25	0.5	0.75	I







		Δ	F		DF	
id	value	6`	L	00	0 0	
А	00		0 1	10	1 1	
В	$01, \pm 10$			v	+	
С	$10, \pm 10$	0	0 0	$1 \perp 1$.0 11	1
D	$10, 1 \bot 1$	F			1	— Lv. 1
Е	, 11, 1⊥1		0		1	
		0	0.25	0.5	0.75	1

		A	k E	e e			
id	value	0		0)(0)	(0)(0)		
А	00		$0 \mid 1$	110	1 1		
В	$01, \pm 10$				+		1/2
С	$10, \pm 10$	0	0 0	$1 \perp 1$.0 1	.1	
D	$10, 1 \bot 1$			+		———————————————————————————————————————	Lv. 1
Е	11, 1⊥1		0		1		
		0	0.25	0.5	0.75	1 1	

		Δ		R	- <u>D</u> E	
id	value	6)		0 0	
A B C	00 01, \perp 10 10, \perp 10	0		$\begin{array}{c} \bot 10 \\ 1 \\ \bot 1 \end{array}$	<u>1⊥1</u> 0 11	Lv. 2 1
D E	$10, 1 \pm 1$ 11, 1 ± 1	·	0		1	— → Lv. 1
MCI	_ = 2 · 3 = 6	0	0.25	0.5	0.75	1

COOL with Binary Encoding



Theoretical Analysis of G-COOL

- Use the Gray code as a fixed encoding in COOL
 - It achieves internal cohesion and external isolation
- Theorem: For the level-k partition \mathscr{C}^k , $x, y \in X$ are in the same cluster if $d_{\infty}(x, y) < 2^{-(k+1)}$
 - Thus x, y are in the different clusters only if $d_{\infty}(x, y) \ge 2^{-(k+1)}$
 - $d_{\infty}(x, y) = \max_{i \in \{1, ..., d\}} |x_i y_i|$ (*L*_{\infty} metric)
 - Two adjacent intervals overlap and they are agglomerated
- Corollary: In the optimal partition \mathcal{C}_{op} , for all $x \in C$ ($C \in \mathcal{C}_{op}$), its nearest neighbor $y \in C$
 - − *y* is nearest neighbor of *x* \iff *y* ∈ argmin_{*y*∈*X*}*d*_∞(*x*, *y*)

Demonstration of G-COOL



Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Experimental Methods

- Analyze G-COOL empirically with synthetic and real datasets compared to DBSCAN and K-means
 - Synthetic datasets were generated by the R package *cluster-Generation* [Qiu and Joe, 2006]

• n = 1,500 for each cluster and d = 3

- Real datasets were geospatial images from Earth-as-Art
 reduced to 200 × 200 pixels, translated into binary images
- All data were normalized by min-max normalization
- G-COOL was implemented by R (version 2.12.1)
- Internal and External measure were used
 - Internal: MCL, connectivity, Silhouette width
 - External: adjusted Rand index



Data show mean \pm s.e.m. Each experiment was performed 20 times











Results (Real datasets)



Results (Real datasets)



Name	n	K	Running time (s)		Running time (s) MC	
			GC	KM	GC	KM
Delta	20748	4	1.158	0.012	4010	4922
Dragon	29826	2	0.595	0.026	3906	7166
Europe	17380	6	2.404	0.041	2320	12210
Norway	22771	5	0.746	0.026	1820	6114
Ganges	18019	6	0.595	0.026	2320	12526

GC: G-COOL, KM: K-means

Outline

- o. Overview
- 1. Background and Our Strategy
- 2. MCL and Clustering
- 3. COOL Algorithm
- 4. G-COOL: COOL with the Gray Code
- 5. Experiments
- 6. Conclusion

Conclusion

- Integrate clustering and its evaluation in the codingoriented manner
 - An effective solution for two essential problems, how to measure goodness of results and how to find good clusters
 No distance calculation and no data distribution
- Key ideas:
 - **1.** *Fix* of an encoding scheme for real-valued variables
 - Introduced the MCL focusing on compression of clusters
 - Formulated clustering with the MCL, and constructed COOL that finds the global optimal solution linearly
 - 2. The Gray code
 - We showed efficiency and effectiveness of G-COOL by theoretically and experimentally

Appendix

Notation (1/2)

- A datum $\mathbf{x} \in \mathbb{R}^d$, a data set $X = \{x_1, \dots, x_n\}$
 - #X is the number of elements in X
 - $X \setminus Y$ is the relative complement of Y in X
- Clustering is partition of X into K subsets (clusters) C_1, \ldots, C_K
 - − $C_i \neq \emptyset$ and $C_i \cap C_j = \emptyset$
 - We call $\mathscr{C} = \{C_1, \dots, C_K\}$ a partition of X
 - $\mathscr{C}(X) = \{\mathscr{C} \mid \mathscr{C} \text{ is a partition of } X\}$
- The set of finite and infinite sequences over an alphabet Σ are denoted by Σ^* and Σ^{ω} , resp.
 - The length |w| is the number of symbols other than \perp
 - If $w = 11 \bot 100 \bot \bot ...$, then |w| = 5
 - For a set of sequences W, $|W| = \sum_{w \in W} |w|$

Notation (2/2)

- An embedding of \mathbb{R}^d is an injective function γ from \mathbb{R}^d to Σ^{ω}
- For $p, q \in \Sigma^{\omega}$, define $p \leq q$ if $p_i = q_i$ for all *i* with $p_i \neq \bot$
 - Intuitively, q is more concrete than p
- For $w \in \Sigma^*$, we write $w \sqsubset p$ if $w \bot^{\omega} \leq p$
 - $\uparrow w = \{p \in range(\gamma) \mid w \sqsubset p\}$ for $w \in \Sigma^*$
 - \uparrow *W* = {*p* ∈ range(*γ*) | *w* ⊏ *p* for some *w* ∈ *W*} for *W* ⊆ Σ^{*}
- The following monotonicity holds

$$-\gamma^{-1}(\uparrow v) \subseteq \gamma^{-1}(\uparrow w) \text{ iff } v \perp^{\omega} \geqslant w \perp^{\omega}$$

Optimization by COOL

- The optimal partition C_{op} can be constructed by the levelk partitions
 - For all $C \in \mathscr{C}_{op}$, there exists k such that $C \in \mathscr{C}^k$
- The level-*k* partitions have the hierarchical structure
 - For each $C \in \mathscr{C}^k$ we have $\bigcup \mathscr{D} = C$ for some $D \subseteq \mathscr{C}^{k+1}$
 - COOL is similar to divisive hierarchical clustering
- COOL always outputs the global optimal partition \mathscr{C}_{op}
- The time complexity is O(nd) (best) and O(nd + K!) (worst)
 - Usually $K \ll n$ holds, hence O(nd)

A-4/A-14
























The Multi-Dimensional Gray Code

- Use the wrapping function $\varphi(p^1, \ldots, p^d) \coloneqq p_1^1 \ldots p_1^d p_2^1 \ldots p_2^d \ldots$
 - Define the *d*-dimensional Gray code embedding $\gamma_G^d: \mathscr{F} \to \Sigma_{\perp,d}^{\omega}$ by $\gamma_G^d(x_1, \ldots, x_d) := \varphi(\gamma_G(x_1), \ldots, \gamma_G(x_d))$
- We abbreviate d of γ_G^d if it is understood from the context

Internal Measures

- Connectivity [Handl et al., 2005]
 - $Conn(\mathcal{C}) = \sum_{x \in X} \sum_{i=1}^{M} f(x, nn(x, i))/i$
 - nn(x, j) is the *i*-th neighbor of x, f(x, y) is 0 if x and y belong to the same cluster, and 1 otherwise
 - *M* is an input parameter (we set as 10)
 - Takes values from 0 to ∞ , should be minimized
- Silhouette width
 - The average of Silhouette value S(x) for each x $S(x) = (b(x) - a(x) / \max(b(x), a(x)))$
 - $a(x) = ||C||^{-1} \sum_{y \in C} d(x, y) \ (x \in C)$
 - $b(x) = \min_{D \in \mathscr{C} \setminus C} \|D\|^{-1} \sum_{y \in D} d(x, y)$
 - Takes values from -1 to 1, should be maximized

A-7/A-14

External Measures

- Adjusted Rand index
 - Let the result be $\mathscr{C} = \{C_1, \dots, C_K\}$ and the correct partition be $\mathscr{D} = \{D_1, \dots, D_M\}$
 - Suppose $n_{ij} \coloneqq ||\{x \in X \mid x \in C_i, x \in D_j\}||$. Then

$$\sum_{i,j \ n_{ij}} C_2 - (\sum_{i \ \|C_i\|} C_2 \sum_{h \ \|D_j\|} C_2) / {}_n C_2$$

$$2^{-1}(\sum_{i \parallel C_i \parallel} C_2 + \sum_{h \parallel D_j \parallel} C_2) - (\sum_{i \parallel C_i \parallel} C_2 \sum_{h \parallel D_j \parallel} C_2)/_n C_2$$

Discussion

- Results for synthetic datasets
 - Best performance under the internal measures
 - (nearly) Best performance under the internal measures
 - G-COOL is efficient and effective
 - DBSCAN is sensitive to input parameters
 - The MCL works well as an internal measure
- Results for real datasets
 - not good, and not bad
 - There are no clear clusters originally
- G-COOL is a good clustering method

Related Work

- Partitional methods [Chaoji et al., 2009]
- Mass-based methods [Ting and Wells, 2010]
- Density-based methods (DBSCAN [Ester et al., 1996])
- Hierarchical clustering methods (CURE [Guha et al., 1998], CHAMELEON [Karypis et al., 1999])
- Grid-based methods (STING [Wang *et al.*, 1997], WaveCluster [Sheikholeslami *et al.*, 1998])

Future Works

- Speeding up by using tree-structures such as BDD
- Apply to anomaly detection
- Theoretical analysis, in particular relation with Computable Analysis
 - Admissibility is a key property

References

[Berkhin, 2006] P. Berkhin. A survey of clustering data mining techniques. Grouping Multidimensional Data, pages 25–71, 2006.

[Chaoji *et al.*, 2009] V. Chaoji, M. A. Hasan, S. Salem, and M. J. Zaki. SPARCL: An effective and efficient algorithm for mining arbitrary shape-based clusters. *Knowledge and Information Systems*, 21(2):201–229, 2009.

[Cilibrasi and Vitányi, 2005] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

- [Ester et al., 1996] M. Ester, H. P. Kriegel, J. Sander, and X. Xu. A densitybased algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of KDD*, 96, 226–231, 1996.
- [Guha *et al.*, 1998] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 1998.

[Handl et al., 2005] J. Handl, J. Knowles, and D. B. Kell. Computational

A-12/A-14

cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201, 2005.

- [Karypis *et al.*, 1999] G. Karypis, H. Eui-Hong, and V. Kumar. CHAMELEON: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.
- [Kontkanen and Myllymäki, 2008] P. Kontkanen and P. Myllymäki. An empirical comparison of NML clustering algorithms. In *Proceedings of Information Theory and Statistical Learning*, 2008.
- [Kontkanen et al., 2005] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I. J. Myung, and M. Pitt, editors, Advances in Minimum Description Length: Theory and Applications. MIT Press, 2005.
- [Knuth, 2005] D. E. Knuth. *The Art of Computer Programming, Volume 4, Fascicle 2: Generating All Tuples and Permutations*. Addison-Wesley Professional, 2005.
- [Qiu and Joe, 2006] W. Qiu and H. Joe. Generation of random clusters with specified degree of separation. *Journal of Classification*, 23:315–334, 2006.

A-13/A-14

[Sheikholeslami *et al.*, 1998] G. Sheikholeslami, S. Chatterjee, and A. Zhang. WaveCluster: A multi-resolution clustering approach for very large spatial databases. In *Proceedings of the 24th International Conference on Very Large Data Bases*, pages 428–439, 1998.

[Ting and Wells, 2010] K. M. Ting and J. R. Wells. Multi-dimensional mass estimation and mass-based clustering. In *Proceedings of 10th IEEE International Conference on Data Mining*, pages 511 – 520, 2010.

[Tsuiki, 2002] Hideki Tsuiki. Real number computation through Gray code embedding. *Theoretical Computer Science*, 284(2):467–485, 2002.

[Wang et al., 1997] W. Wang, J. Yang, and R. Muntz. STING: A statistical information grid approach to spatial data mining. In *Proceedings* of the 23rd International Conference on Very Large Data Bases, pages 186–195, 1997.

[Weihrauch, 2000] K. Weihrauch. *Computable Analysis*. Springer, 2000.