Constraint Programming for Constrained Clustering

C. Vrain

LIFO - Université d'Orléans - France

FCA4AI 2016

Data Mining from the point of view of search

Two families of problems

- Enumeration: pattern mining
  - Enumerating frequent itemsets
  - Enumerating closed frequent itemsets
  - ...
  - Extended to more complex data structures: sequences, graphs, ...
- Optimization:
  - Supervised Classification
  - Clustering: unsupervised learning, only modeled by the optimization criterion

$\rightarrow$ Exact methods / Approximated methods
$\rightarrow$ Global optimum / local optimum

Constrained Data Mining

Introduction of constraints

- To reduce the number of patterns and to find more interesting patterns (enumeration problems)
- To better fit the needs of the user, only modeled in the optimization criterion (optimization problems)

$\rightarrow$ Different kinds of constraints

$\rightarrow$ Needs to adapt classic algorithms to handle these constraints

$\rightarrow$ Algorithms designed to handle some types of constraints

A new research domain: Declarative Frameworks for Data Mining

Declarative framework for Data Mining

- Frequent itemset mining
  - First work of *(L. de Raedt & al, KDD 08)* on frequent itemset mining in Constraint Programming (CP)
  - Extension to k-pattern-set mining *(Khiari & al. CP 2010, Guns & al., 2010)* $\rightarrow$ Application to conceptual clustering
  - A global constraint for closed itemset mining *(Lazaar & al. CP 2016)*
- Clustering
  - Conceptual clustering in an Integer LP framework *(Mueller & al., 2010)*.
  - Constrained distance-based clustering
    - in SAT (2 classes) *(Davidson & al, 2010)*
    - in CP (k classes - k bounded) in CP *(Dao, Duong & Vrain, ECML/PKDD 2013, CP 2014, AIJ 2015)*
    - in an Integer LP framework *(Babaki et al., 2014)*
- Sequence mining: *(Kemmar & al., CP 2015)*, *(Négrevergne & al., CPAIOR 2015)*, *(Aoga & al. ECML/PKDD 2016)*, *(Gelser & al. IJCAI 2016)*

The list is not exhaustive !

Outline

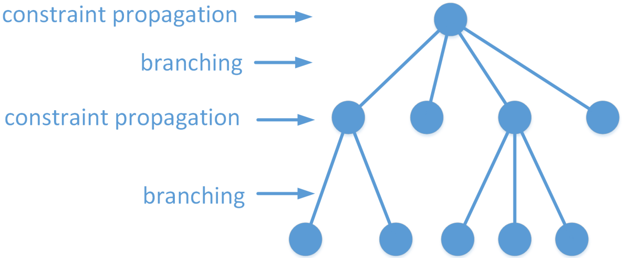
1. Constraint Programming
2. Pattern mining in CP
3. Distance Based Clustering
   - Constrained Clustering
   - Distance-based constrained clustering in CP: How-to? *(Dao, Duong, Vrain)*
4. Interest of declarativity *(Dao, Duong, Vrain)*
   - Flexibility
   - Finding a global optimum
   - Embedding in other algorithms
   - Integration in a more general framework
5. Conclusion

Outline

1. Constraint Programming
2. Pattern mining in CP
3. Distance Based Clustering
   - Constrained Clustering
   - Distance-based constrained clustering in CP: How-to? *(Dao, Duong, Vrain)*
4. Interest of declarativity *(Dao, Duong, Vrain)*
   - Flexibility
   - Finding a global optimum
   - Embedding in other algorithms
   - Integration in a more general framework
5. Conclusion

Constraint Programming (CP)

- Declarative Modeling of the problem by specifying variables and constraints
- Search for solutions by constraint propagation and branching



For optimization

- *branch-and-bound* strategy for optimizing a criterion
  Once a solution is found $\rightarrow$ add constraints to forbid less good solutions

Let Δ a solution found, *obj* the objective to minimize and *OBJ* the variable capturing it.

- computation of $obj(\Delta)$
- addition of the constraint $OBJ < D(\Delta)$

An example in CP

$$SEND + MOST = MONEY$$

- Find all the assignments of digits to letters such that

$$\begin{array}{ccccc} & S & E & N & D \\ + & M & O & S & T \\ \hline M & O & N & E & Y \end{array}$$

$\rightarrow$ CSP : constraint satisfaction problem

- Find an assignment of digits to letters such that
  - $SEND + MOST = MONEY$
  - *MONEY* is maximal

$\rightarrow$ COP: constraint optimization problem

Modeling in CP

$$SEND + MOST = MONEY$$

- Variables $S, E, N, D, M, O, T, Y \in \{0, \ldots, 9\}$
- Variable V capturing the value of the objective function
- Constraints:
  - $S \neq 0, \; M \neq 0$
  - `alldifferent`(S, E, N, D, M, O, T, Y)
  - A linear constraint

$$\begin{array}{rc} & (1000 \times S + 100 \times E + 10 \times N + D) \\ + & (1000 \times M + 100 \times 0 + 10 \times S + T) \\ = & 10000 \times M + 1000 \times 0 + 100 \times N + 10 \times E + Y \end{array}$$

  - $V = 10000 \times M + 1000 \times 0 + 100 \times N + 10 \times E + Y$
- Objective function: *maximize* V

Solvers

- CP solvers: iteration of
  - Constraint propagation: removing inconsistent values from the variable domains

$$
\begin{aligned}
&D_S = \{9\}\\
&D_E = \{2,3,4,5,6,7\}\\
&D_M = \{1\}\\
&D_O = \{0\}\\
&D_N = \{3,4,5,6,7,8\}\\
&D_D = D_T = D_Y = \{2,3,4,5,6,7,8\}
\end{aligned}
$$

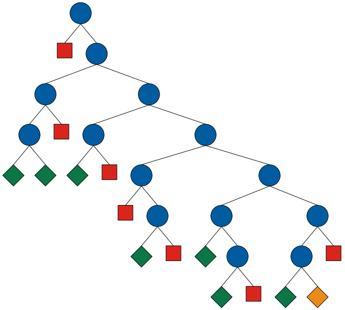
  - Branching: creation of branches in the search tree

$$
\text{Binary tree: } E = 2 \text{ and } E \neq 2
$$
$$
\rightarrow
$$
$$
D_E = \{2\} \quad \text{and} \quad D_E = \{3,4,5,6,7\}
$$

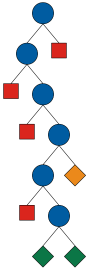
  - In case of optimization, add a new constraint each time a solution is found

Influence of search strategies

S, E, N, D, M, O, T, Y



S, T, Y, N, D, E, M, O



stable state
failure state
intermediary solution
best solution

Global Constraints

Global constraints

Constraints embedding a set of constraints
$\Rightarrow$ more powerful filtering algorithms

Pairwise distinct values

- Elementary Constraints: $S \neq E, S \neq N, E \neq N, \ldots$
  - $S \in \{1,2,3\}, E \in \{1,2\}, N \in \{1,2\}$

  $\Rightarrow$ $S \in \{1,2,3\}$, $E \in \{1,2\}, N \in \{1,2\}$
- A global constraint: *alldifferent*(S, E, N, D, M, O, T, Y)
  - $S \in \{1,2,3\}, E \in \{1,2\}, N \in \{1,2\}$

  $\Rightarrow$ $S = 3$, $E \in \{1,2\}, N \in \{1,2\}$

Outline

Pattern Mining

$\mathcal{O}$: a set of n objects described by m boolean properties ($\mathcal{I}$)
$\mathcal{D}$: a data matrix

	a	b	c
o_1	1	1	0
o_2	1	1	1
o_3	0	1	1
o_4	0	1	1
o_5	0	1	1

- pattern: an itemset $p_1, \ldots, p_j$, closed when all objects satisfying $p_1, \ldots p_j$ have only these objects in common.
- concept: (a set of objects, an itemset) such that these objects, and only them, satisfy the set of items.

$$(\{o_2\}, \{a, b, c\}), (\{o_1, o_2\}, \{a, b\}), (\{o_2, o_3, o_4, o_5\}, \{b, c\})$$

Itemset mining in CP *(de Raedt & al. 2008)*

- Inputs: $(\mathcal{O}, \mathcal{I}, \mathcal{D})$: $\forall o \in \mathcal{O}, p \in \mathcal{I}$, $D_{oa} \in \{0, 1\}$
- A concept π is defined by:
  - a set of objects

  $\rightarrow$ n boolean variables T_i: true when data i belongs to the concept
  - a set of properties

  $\rightarrow$ m boolean variables I_j: true when property j is satisfied in the concept

	a	b	c
o_1	1	1	0
o_2	1	1	1
o_3	0	1	1
o_4	0	1	1
o_5	0	1	1

$(\{o_2\}, \{a, b, c\})$

T_1	T_2	T_3	T_4	T_5
0	1	0	0	0

I_1	I_2	I_3
1	1	1

$(\{o_1, o_2\}, \{a, b, \})$

T_1	T_2	T_3	T_4	T_5
1	1	0	0	0

I_1	I_2	I_3
1	1	0

Itemset mining in CP *(de Raedt & al. 2008)*

- Frequent itemset π

Extension constraint (*coverage*) $T = I'$ + Frequency constraint

$$\forall o \in \mathcal{O} \qquad T_o = 1 \leftrightarrow \sum_{p \in \mathcal{I}} I_p \times (1 - D_{op}) = 0$$
$$\sum_{o \in \mathcal{O}} T_o \geq \theta$$
$$\forall p \in \mathcal{I} \qquad I_p = 1 \rightarrow \sum_{o \in \mathcal{O}} I_p \times D_{op} \geq \theta$$

- Closed itemset π

Intension constraint (*closed*) $I = T'$

$$\forall p \in \mathcal{I} \qquad I_p = 1 \leftrightarrow \sum_{o \in \mathcal{O}} T_o(1 - D_{op}) = 0$$

- Other constraints for specifying maximal itemset mining, maximum total cost constraint, emerging patterns ...

***size_pattern*$(\pi) \geq \theta$, *tile*$(\pi) \leq \theta$**

$$\sum_{p \in \mathcal{I}} I_p \geq \theta$$
$$\left(\sum_{p \in \mathcal{I}} I_p\right) \times \left(\sum_{o \in \mathcal{O}} T_o\right) \leq \theta$$

k-pattern mining in CP *(Guns & al.)*

- Search for k patterns $\Pi = (\pi_1, \ldots, \pi_k)$ (thus defining k concepts) that satisfy a set of constraints
- Application to k-term DNF Learning ($\mathcal{O}^+$,$\mathcal{O}^-$), k-tiling, conceptual clustering, ...

Conceptual clustering

- $\forall \pi_i$, $covers(\pi_i)$
- $\forall \pi_i$, $closed(\pi_i)$
- $cover(\Pi) = coverage(\pi_1) \cup \ldots \cup coverage(\pi_k) = \mathcal{O}$
- $\forall \pi_i$, $\forall \pi_j$ with $i \neq j$, $overlap(\pi_i, \pi_j) = 0$

maximizing

- $min(freq(\pi_1), \ldots, freq(\pi_k))$ or
- $max(freq(\pi_1), \ldots, freq(\pi_k)) - min(freq(\pi_1), \ldots, freq(\pi_k))$

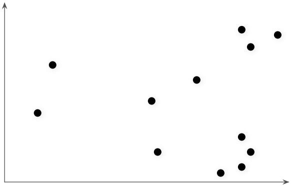
Outline

Outline

1. Constraint Programming
2. Pattern mining in CP
3. Distance Based Clustering
   - Constrained Clustering
   - Distance-based constrained clustering in CP: How-to? *(Dao, Duong, Vrain)*
4. Interest of declarativity *(Dao, Duong, Vrain)*
   - Flexibility
   - Finding a global optimum
   - Embedding in other algorithms
   - Integration in a more general framework
5. Conclusion

Clustering

Given n objects $\{o_1, \ldots, o_n\}$, find a partition of these objects into k classes so that objects in a class are similar and/or objects of different classes are dissimilar.



Three approaches

- Distance-based clustering: a dissimilarity measure between pairs of points
- Conceptual clustering
- Correlation clustering: a similarity between pairs of points

Distance-based Clustering

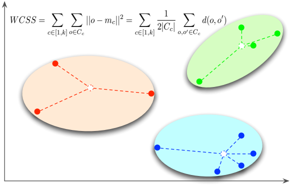
A classic example: k-means

Minimize the *within-cluster sum of squares* (WCSS) where if m_c denotes the center of the cluster C_c,

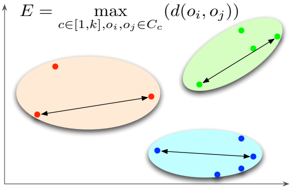
$$WCSS(\Delta) = \sum_{c \in [1,k]} \sum_{o_i \in C_c} d(o_i, m_c)^2$$

equivalent in an euclidean space to minimize

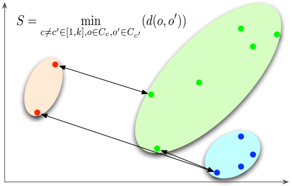
$$WCSS(\Delta) = 1/2 \sum_{c \in [1,k]} \frac{1}{|C_c|} \sum_{o_i, o_j \in C_c} d(o_i, o_j)^2.$$



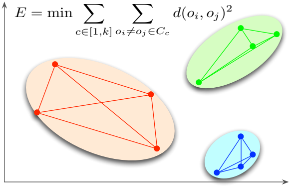
Optimization criteria



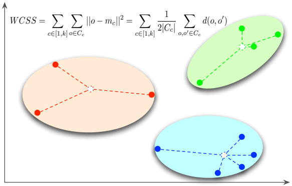
Minimization of the maximal diameter $d(o_i, o_j)$
(polynomial for k=2)



Maximization of the minimal margin
(polynomial)



Minimization of the sum of dissimilarities
WCSD



Minimization of the sum of squares
WCSS

Few exact methods for clustering

- Methods based on graphs
  - Minimization of the maximum diameter: graph coloring *(Hansen & Delattre, 1978)*
- Branch and bound algorithms
  - Diameter criterion *(Brusco 2003)*
  - WCSD criterion *(Klein & Aronson 91, Brusco & Stahl 2005)*
  - WCSS criterion *(Koontz 1975, Brusco 2006, Carbonneau & al. 2012)*
- Integer Linear Programming *(Rao, 79, du Merle & al. 1999, Mueller & al. 2010, Babaki & al. 2014)*

Constrained Clustering

- Clustering is in general NP-hard.

- Classic methods are usually heuristic and search for a local optimum. Different local optima may exist.

- The result does not always fit the desired solution.

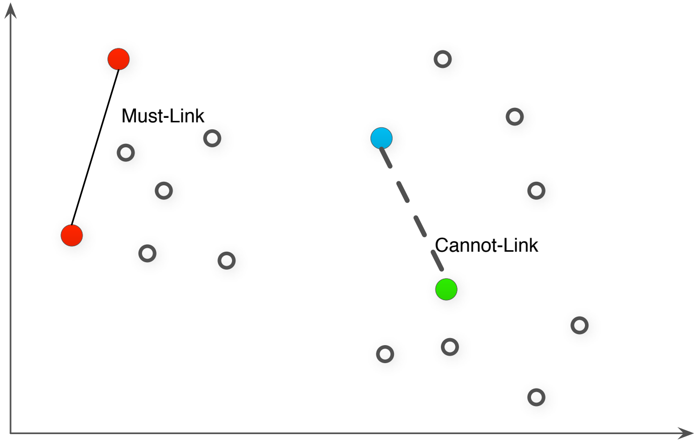
$\rightarrow$ User knowledge integration, written as constraints

$\rightarrow$ Constrained clustering

  - Constraints on clusters
  - Constraints on pairs of points

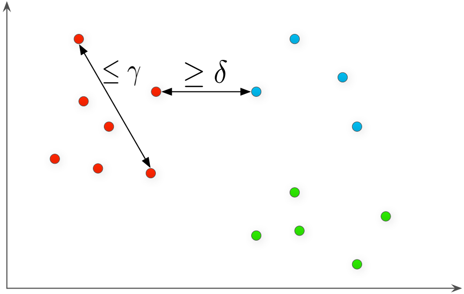
$\rightarrow$ Mostly heuristic methods dedicated to some types of constraints

Constraints on pairs of points



Constraints on clusters

- Capacity constraint:
  $\alpha \leq |C_i| \leq \beta$
- Maximal diameter constraint
- Minimal margin constraint
- Density constraint
- ...



Outline

Distance-based clustering

Inputs:

- a dissimilarity measure between pairs of objects
- a bound on the number of classes: $k_{min} \leq k \leq k_{max}$

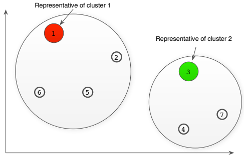
Output: A data partition

- integration of different kinds of constraints
- choice of an optimization criterion among :
  - Minimizing the maximal diameter of clusters
  - Maximizing the minimal split between clusters
  - Minimizing the sum of dissimilarities inside the clusters

(Dao, Duong & Vrain, ECML/PKDD 2013, ICTAI 2013, RIA 2014, CP 2015, AIJ 2015)

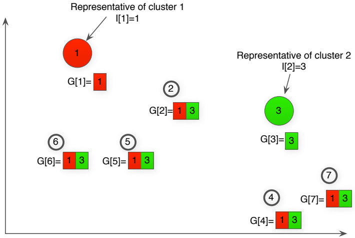
First model *(Dao, Duong & Vrain, ECML/PKDD 2013)*

- A cluster is identified by a representative
- k variables: $I[1], \ldots, I[k] \in [1, n]$
  $I[c]$: representative point of the cluster c



- For breaking symmetries: the point with the smallest index in the cluster

- Each point is linked to a point representing a cluster
- n variables $G[1], \ldots, G[n] \in [1, n]$
  $G[i]$: representative point of the cluster containing i



Second model *(Dao, Duong & Vrain, AIJ 2015)*

- n integer variables: $G_1, \ldots, G_n$, $dom(G_i) = \{1, \ldots, k_{max}\}$.

$G_1 = 1, G_2 \in [1,2], G_3 \in [1,2]$
Point 1 is put in Cluster 1, Point 2 may still be put in Cluster 1 or 2 and so on

$G_i = c$: point i is assigned to class c

- A real variable capturing the objective to optimize:
  - D: maximal diameter
  - S: minimal split
  - V: sum of intra-cluster dissimilarities

  - $dom(D) = dom(S) = [\min_{i,j}(d(i,j)), \max_{i,j}(d(i,j))]$
  - $dom(V) = [0, \sum_{i<j} d(i,j)]$.

Constraints on the partition

- Breaking symmetries:

precede constraint:
$$G_1 = 1 \text{ and } G_i \leq \max_{j\in[1,i-1]}(G_j) + 1, \text{ for } i \in [2, n]$$

- $precede(\mathcal{G}, [1, \ldots, k_{max}])$
  means $G_1 = 1$ and $\forall i \in [2, n]$, if $G_i = c$ then $\exists j < i\ G_j = c - 1$

- At least k_{min} clusters:

count constraint: $\#\{i \mid G_i = k_{min}\} \geq 1$

- $atleast(1, \mathcal{G}, k_{min})$

Adding redundant constraints may help filtering.

User constraints

A minimal size α for clusters: $\forall i \in [1, n], \#\{j \mid G_j = G_i\} \geq \alpha$

- $\forall i \in [1, n]$: $atleast(\alpha, \mathcal{G}, G_i)$
- $G_i \leq \lfloor n/\alpha \rfloor, \forall i \in [1, n]$

A maximal diameter γ for clusters:

- $D \leq \gamma$
- $\forall i, j$ such that $d(i, j) > \gamma$, the constraint $G_i \neq G_j$ is put

Constraints on pairs of points

A must-link constraint: $G_i = G_j$ et $D \geq d(i, j)$
A cannot-link constraint: $G_i \neq G_j$ et $S \leq d(i, j)$

Search strategy

1. Initial ordering of points $\rightarrow$ use of FPF
2. Strategy for choosing variables

Optimization of WCSD

- greedy search for "finding" quickly a "good" first solution
- then strategy change for detecting quickly failure

3. Development of filtering algorithms

Filtering algorithms

1. To improve the efficiency

Diameter constraint

$\forall i < j \in [1, n]$
$D < d(i, j) \rightarrow G_i \neq G_j$
Implementation by reified constraints
A quadratic number of reified constraints

$\rightarrow$ a dedicated filtering algorithm

2. To improve the filtering capacity

$WCSD : V = \Sigma_{i<j}(G_i == G_j)d(i, j)$

development of an algorithm to improve the clustering

Not enough propagation for WCSD

- Let us suppose that we have a solution with $V = 5$, the branch-and-bound strategy add a new constraint

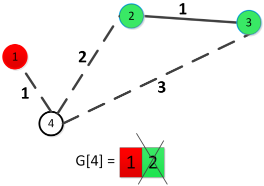
$$(G_1 == G_4) + 2 \times (G_2 == G_4) + 3 \times (G_3 == G_4) + 1 < 5$$

  - Points 2 et 3 are in the same cluster

  $\rightarrow$ $(G_2 == G_4) = (G_3 == G_4)$

  - If $(G_2 == G_4)$ then $(G_3 == G_4)$ and the sum is greater to 5

  $\rightarrow$ Point 4 cannot be in cluster 2



- But value 2 is not removed from the domain of G_4

$\rightarrow$ A filtering algorithm *(Dao, Duong, Vrain, ICTAI 2013)*

Two models

	First model	Second model
Var.	$\mathcal{I} = [I_1, \ldots, I_k]$, $Dom(I_c) = [1, n]$ $\mathcal{G} = [G_1, \ldots, G_n]$, $Dom(G_i) = [1, n]$	$\mathcal{G} = [G_1, \ldots, G_n]$, $Dom(G_i) = [1, k_{max}]$
	D (diameter), S (split), V (WCSD)	
Partition	$\forall c \in [1, k]$, $element(\mathcal{G}, I_c, I_c)$ $\forall i \in [1, n]$, $exactly(1, \mathcal{I}, G_i)$ $\forall i \in [1, n]$, $G_i \leq i$ $\forall c < c' \in [1, k]$, $I_c \leq I_{c'}$ $I_1 = 1$	$precede(\mathcal{G}, [1, \ldots, k_{max}])$ $atleast(1, \mathcal{G}, k_{min})$
Opt. Crit.	WCSD criterion: $wcsd(\mathcal{G}, V, d)$	
	Diameter criterion:	
	$\forall i < j \in [1, n]$, $d(i, j) > D \rightarrow (G_i \neq G_j)$	$diameter(\mathcal{G}, D, d)$
	Split criterion:	
	$\forall i < j \in [1, n]$, $d(i, j) < S \rightarrow (G_i = G_j)$	$split(\mathcal{G}, S, d)$

Two models

	First model	Second model
User-constraints	Minimal size α of clusters: $\forall i \in [1, n]$, *atleast*$(\alpha, \mathcal{G}, G_i)$	
	Maximal size β of clusters: $\forall c \in [1, k_{max}]$, *atmost*$(\beta, \mathcal{G}, c)$	
	Minimal split δ: $S \geq \delta$, $G_i = G_j$, for all $i < j$ st. $d(i,j) < \delta$	
	Maximal diameter γ: $D \leq \gamma$, $G_i \neq G_j$ for all $i < j$ st. $d(i,j) > \gamma$	
	Density constraint: $\forall i \in [1, n]$, *atleast*(m, N_{i_ϵ}, G_i)	
	Must-link constraint: $G_i = G_j$, $D \geq d(i,j)$	
	Cannot-link constraint: $G_i \neq G_j$, $S \leq d(i,j)$	

Experiments

→ choice of the models and of the strategies

Dataset	# Objets	# Classes
Iris	150	3
Wine	178	3
Glass	214	7
Ionosphere	351	2
User Knowledge	403	4
Breast Cancer	569	2
Synthetic Control	600	6
Vehicle	846	4
Yeast	1484	10
Multiple Features	2000	10
Image Segmentation	2000	7
Waveform	5000	3

Experiments

- Minimization of the maximal diameter
- Comparison between
  - BaB: *branch-and-bound* approach *(Brusco 2005)*
  - GC: graph coloring *(Hansen 1980)*
  - CP1: first model *(Dao, Duong & Vrain, ECML 2013, ICTAI2013)*
  - CP2: second model *(Dao, Duong & Vrain, IAF 2015)*
- Without user constraints

Experiments

Dataset	D_{opt}	BaB	GC	CP1	CP2
Iris	2.58	1.4	1.8	< 0.1	< 0.1
Wine	458.13	2	2.3	0.3	< 0.1
Glass	4.97	8.1	42	0.9	0.2
IonoSphere	8.6	–	0.6	0.4	0.3
User Knowledge	1.17	–	3.7	75	0.2
Breast Cancer	2377.96	–	1.8	0.7	0.5
Synthetic Control	109.36	–	–	56.1	1.6
Vehicle	264.83	–	–	14.3	0.9
Yeast	0.67	–	–	2389.9	5.2
Multi Features	12505.5	–	–	$*$	10.4
Image Segmentation	436.4	–	–	589.2	5.7
Waveform	15.6	–	–	$*$	50.1

Performance (in seconds) - minimization of the maximal diameter

And the WCSS criterion? (k-means)

$$WCSS = \sum_{c \in [1,k]} \frac{1}{2|C_c|} \sum_{o_i, o_j \in C_c} ||o_i - o_j||^2$$

- RBBA: Repetitive Branch and Bound Algorithm *(Brusco, 2006)*, without user constraints
  - Points ordering
  - Iteratively, solve the problem for $k+1, \ldots, k+n$ objects: the optimal value at one step gives a bound for the other step
  - Experiments:
    - well separated clusters: n=240, k=8
    - no underlying structure: n=60, k=6
- Integer Linear Programming with user constraints: *(Babaki & al. 2014)*

And the WCSS criterion?

1. Proposition of a filtering algorithm for WCSS *(Dao, Duong, Vrain, CP 2015)*
2.

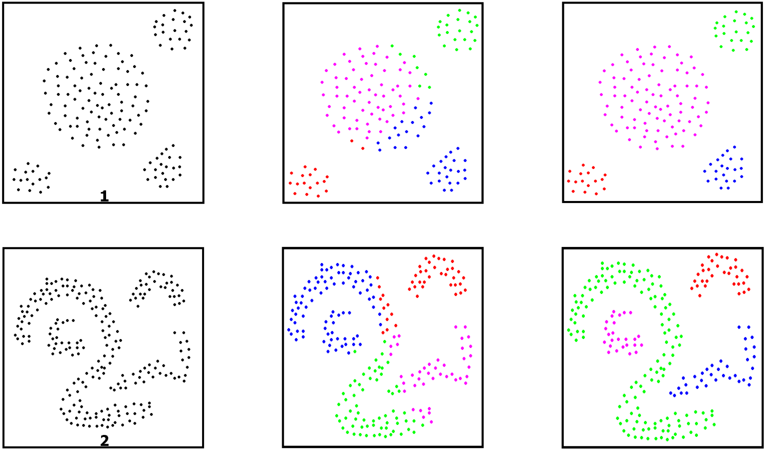
See presentation on Friday morning
Repetitive Branch-and-Bound using Constraint Programming for Constrained MSS Clustering
T. Guns, T.-B.-H. Dao, C. Vrain, K.-C. Duong
Friday, 10:30 – 12:10 Yangtze 2

Outline

1. Constraint Programming
2. Pattern mining in CP
3. Distance Based Clustering
   - Constrained Clustering
   - Distance-based constrained clustering in CP: How-to? *(Dao, Duong, Vrain)*
4. Interest of declarativity *(Dao, Duong, Vrain)*
   - Flexibility
   - Finding a global optimum
   - Embedding in other algorithms
   - Integration in a more general framework
5. Conclusion

Outline

Advantage of Flexibility

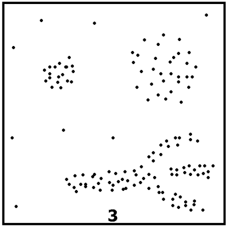


Dataset

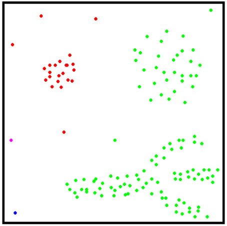
Diameter criterion

Diameter criterion +
margin constraint

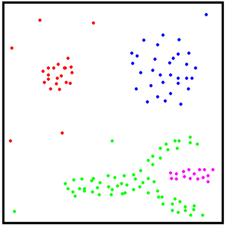
Advantage of Flexibility



Dataset



Margin criterion



Margin criterion + density constraint

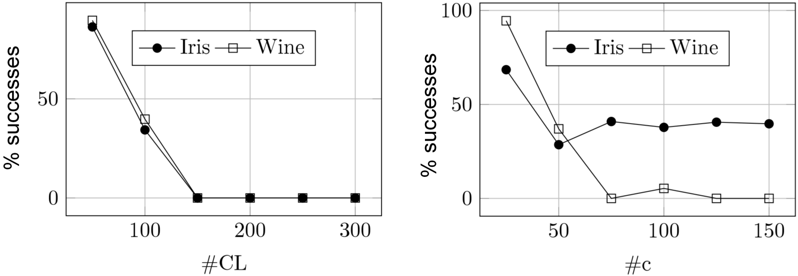
Human in the loop

$\rightarrow$ suggest an iterative process: Search - Adapt/Learn constraints

Outline

Importance of exact algorithms

- COP-kmeans is a fast and greedy algorithm, which may fail when the number of cannot-link constraints increases
- Dataset Iris, COP-kmeans runs 1000 times
  left: COP-kmeans, $\#CL$ cannot-link constraints
  right: COP-kmeans, $\#c$ must-link and $\#c$ cannot-link constraints



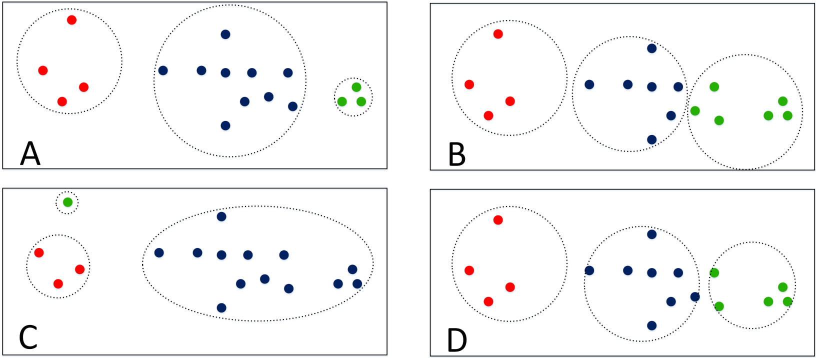
In these experiments, our CP model

- always finds a solution satisfying all the constraints,
- for Iris dataset, succeeds in proving the optimality for roughly 60% cases.

Outline

Bi-criterion clustering

Criteria have often undesirable effect



(A) Intuitive groups (B) Minimizing max diameter
(C) Maximizing min split (D) Miminizing WCSS

Bi-criterion split-diameter clustering

$$(\min D, \max S)$$

$\rightarrow$ find the Pareto front

- A partition Δ' dominates a partition Δ iff:
  $D(\Delta') \leq D(\Delta)$ and $S(\Delta') > S(\Delta)$ or
  $D(\Delta') < D(\Delta)$ and $S(\Delta') \geq S(\Delta)$.
- Δ is a Pareto-optimal solution iff there exists no partition Δ' that dominates Δ
- Pareto front = $\{(D(P), S(P)) \mid P \text{Pareto-optimal solution}\}$

Split-diameter clustering

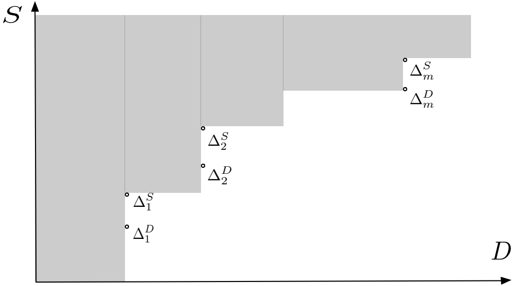
(Dao, Duong, Vrain, AIJ 2015)

- Iteration of the model by adding constraints

$\rightarrow$ interest of a declarative framework

Algorithm

$\mathcal{A} \leftarrow \emptyset$;
$i \leftarrow 1$;
$\Delta_i^D \leftarrow$ Min_Diameter($\mathcal{C}$);
while $\Delta_i^D \neq NULL$ **do**
  $\Delta_i^S \leftarrow$ Max_Split($\mathcal{C} \cup \{D \leq D(\Delta_i^D)\}$);
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{\Delta_i^S\}$;
  $i \leftarrow i + 1$;
  $\Delta_i^D \leftarrow$ Min_Diameter($\mathcal{C} \cup \{S > S(\Delta_{i-1}^S)\}$);



Experiments

split-diameter bi-criterion - $k \in [2, \text{real number of classes}]$

Dataset	n	k	#Sol	bGC	CP2
Iris	150	3	8	4.2	< 0.1
Wine	178	3	8	0.9	< 0.1
Glass	214	7	9	21.5	0.4
Ionosphere	351	2	6	1.8	2.6
User Knowledge	403	4	16	23.6	12.8
Breast Cancer	569	2	7	167.5	1.1
Synthetic Control	600	6	6	–	6.7
Vehicle	846	4	13	–	5.5
Yeast	1484	10	–	–	–
Multi Features	2000	10	15	–	229.1
Image Segmentation	2000	7	8	–	41.3
Waveform	5000	3	–	–	–

- bGC: best exact algorithm known (Delattre *et al.*, 1980), coded in C++
- Times in seconds, time out 1 hour

Outline

Two frameworks for clustering in CP

- Conceptual Clustering *(Guns & al.)*:
  - based on k-pattern mining
  - based on qualitative properties
  - does not take into account quantitative information, e.g. clusters diameter
- Distance-based clustering *(Dao & al.)*:
  - based on dissimilarities between objects
  - appropriate for quantitative data
  - does not take into account qualitative properties

$\rightarrow$ A unified framework *(Dao, Lesaint, Vrain, JFPC 2015)*:
  - taking into account quantitative and qualitative data
  - modeling each clustering task
  - combining conditions/criteria from both frameworks
  - relying on the framework developed for distance-based clustering

An integrated model

- Data:
  - a set $\mathcal{O}$ of objects, a set $\mathcal{I}$ of boolean properties
  - a dissimilarity measure $d(o, o')$ for any pairs of objects o, o' in $\mathcal{O}$ or a database from which the dissimilarity measure is computed
  - a binary database $\mathcal{D}$: $\forall o \in \mathcal{O}, p \in \mathcal{I}$, $D_{op} = 1$, when o satisfies p
- Clusters are defined by:
  1. n variables with dom in $[1, k]$: $G[o] = c$
  2. a boolean $k \times n$ matrix: $A[c, p] = 1$ iff p is in the description of cluster c.
- Constraints
  - Constraints of the distance-based model: partition, breaking symmetries
  - Constraints from the conceptual model

Extension (cover) constraint

$\forall o \in \mathcal{O}, \forall c \in \mathcal{C}$

$$G[o] = c \Leftrightarrow \sum_{p \in \mathcal{I}} A[c, p](1 - D_{op}) = 0$$

Car Dataset

- 193 objects
- technical properties (22 attributes) :
  - motorization (diesel or not)
  - drive wheels (4, 2 front, 2 rear)
  - power (between 48 and 288)
  - etc.

  discretization : 64 qualitative attributes
- price (quantitative attribute)

Car dataset

- Conceptual setting

→ (e) concepts + maximizing min. size of clusters

→ (f) concepts + maximizing min. size of concepts

Price distribution not convincing

- Distance-based setting

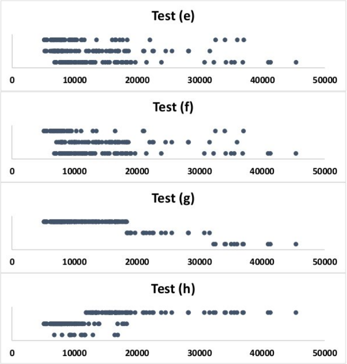
→ (g) minimizing max diameter

No convincing concepts

- Unified framework

→ (h) concepts + minimizing max diameter

A better modeling of the 3 car ranges with concepts based on size, engine power, fuel consumption, . . .



Actionable clustering

Find useful groups each of which you can invite to a different dinner party

- equal number of males and females
- width of a cluster in terms of age at most 10
- each person in a cluster should have at least *r* other people with the same hobby

$\rightarrow$ Introduce requirements/constraints making the clustering useful for a given purpose.

See presentation on Friday morning
A Framework for Actionable Clustering Using Constraint Programming
T.-B.-H. Dao, C. Vrain, I. Davidson, K.-C. Duong
Friday, 10:30 – 12:10 Yangtze 2

Outline

Conclusion

Declarative frameworks for constrained clustering, and more generally for Data Mining

CP, Integer Linear Programming, SAT

Advantages

- Handling different kinds of constraints
- → A better modeling of user needs
- → New paradigms: *actionable clustering*
- Exact methods
- Integration in other frameworks
- → Bi-criterion clustering
- → Embedding in other search methods

Drawbacks

- Efficiency
- → Small / medium size datasets
- → Smart data
- Design of an efficient model difficult
- In CP
- → Constraints allowing to filter domains / Constraints only tested at the bottom of the tree
- → Combining several paradigms
- → Parallelization: Enumeration / Optimization

Further directions

- User in the loop
  - Feedbacks for guiding the user
  - Learning constraints
- Combining several frameworks
- Improving the efficiency

Thanks to T.B.H. Dao and K.C. Duong
for some slides and figures