Des méthodes déclaratives exactes pour la classification non supervisée sous contraintes

Christel VRAIN
en collaboration avec
Thi-Bich-Hanh DAO et Khanh-Chuong DUONG

LIFO
Université d'Orléans

Atelier CluCo, EGC 2015

Plan

Plan

Classification non supervisée

Etant donnés n objets $\{o_1, \ldots, o_n\}$, trouver une partition de ces objets en k classes telle que les objets d'une même classe soient similaires et/ou les objets de deux classes différentes soient dissimilaires.

$\rightarrow$ Minimiser la somme des carrés intra-cluster (WCSS *within-cluster sum of squares* :
si m_c est le centre du cluster C_c,

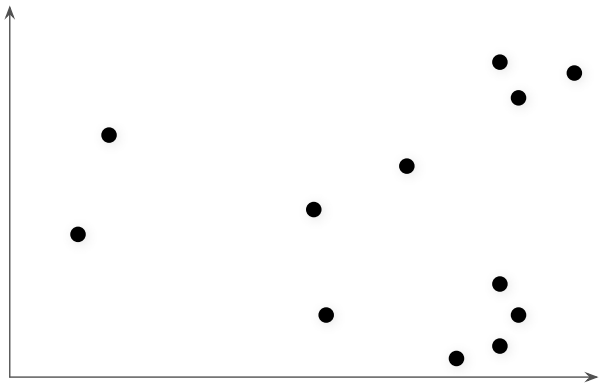
$$WCSS(\Delta) = \sum_{c \in [1,k]} \sum_{o_i \in C_c} d(o_i, m_c)^2$$

équivalente dans un espace euclidien à

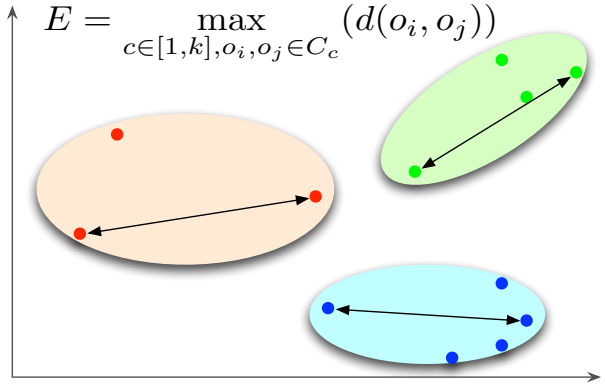
$$WCSS(\Delta) = \sum_{c \in [1,k]} \frac{1}{|C_c|} \sum_{o_i, o_j \in C_c} d(o_i, o_j)^2.$$

$\rightarrow$ Mais il existe d'autres critères

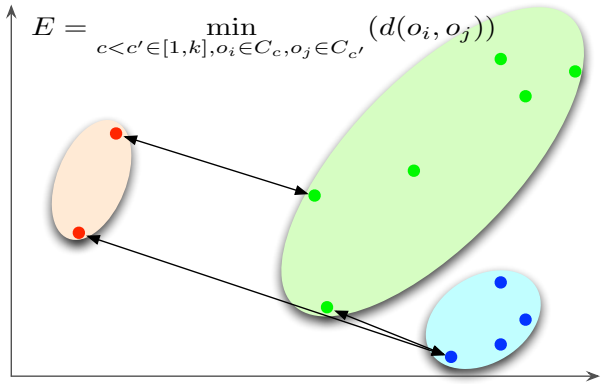
Classification non supervisée



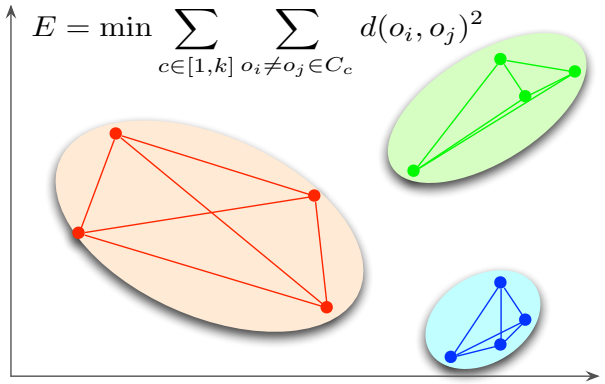
Dissimilarité $d(o_i, o_j)$



Minimisation du diamètre maximal
(polynomial pour k=2)



Maximisation de la marge minimale
(polynomial)



Minimisation de la somme des dissimilarités
WCSD

Peu de méthodes exactes

- Méthodes fondées sur les graphes
  - minimisation du diamètre maximum : coloriage de graphe *(Hansen & Delattre, 1978)*
- Algorithmes de type "Branch and bound"
  - critère du diamètre *(Brusco 2003)*
  - critère WCSD *(Klein & Aronson 91, Brusco & Stahl 2005)*
  - critère WCSS *(Koontz 1975, Brusco 2006, Carbonneau & al. 2012)*
- Integer Linear Programming *(Rao, 79, du Merle & al. 1999, Aloise & al. 2009)*

$\mathcal{C}$: *ens. des* 2^n *clusters*

$a_{it} = 1$ *si* $o_i \in C_t$

c_t : *coût du cluster* t, $c_t = \Sigma_{i=1}^{n} d(o_i, m_t)^2 a_{it}$

$$\begin{aligned} &\textit{minimiser}_x \Sigma_{t\in\mathcal{C}} c_t x_t \\ &\Sigma_{t\in\mathcal{C}} x_t a_{it} = 1, \forall i \in [1..n] \\ &\Sigma_{t\in\mathcal{C}} x_t = k \\ &x_t \in \{0,1\}, \forall t \in \mathcal{C} \end{aligned}$$

Clustering avec contraintes

- Le clustering est en général NP-difficile.
- Les méthodes classiques sont heuristiques et cherchent un optimum local. Différents optima locaux peuvent exister.

$\rightarrow$ Intégration de connaissances sous forme de contraintes

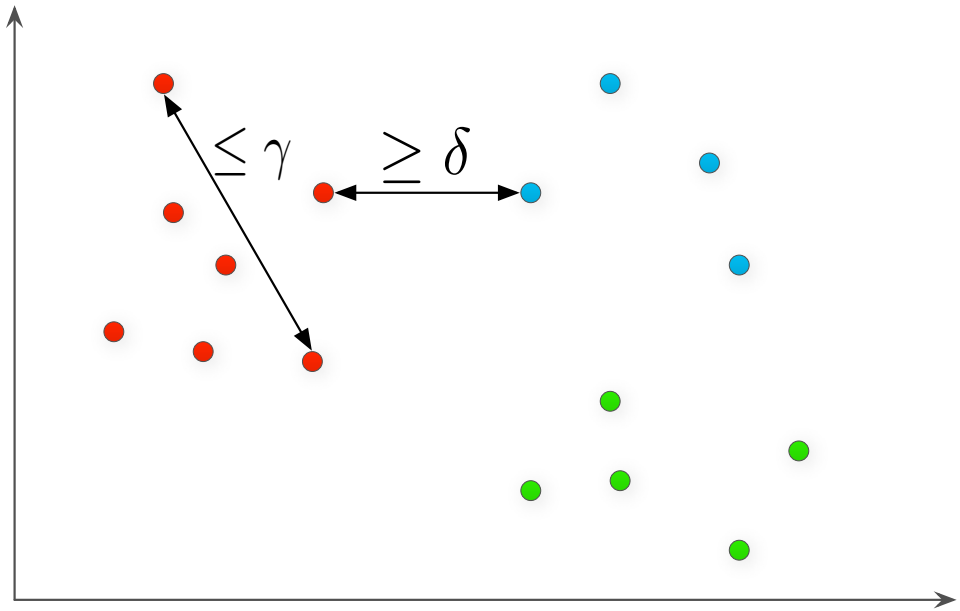
$\rightarrow$ Clustering sous contraintes

- Contraintes sur les clusters
- Contraintes sur les paires de points

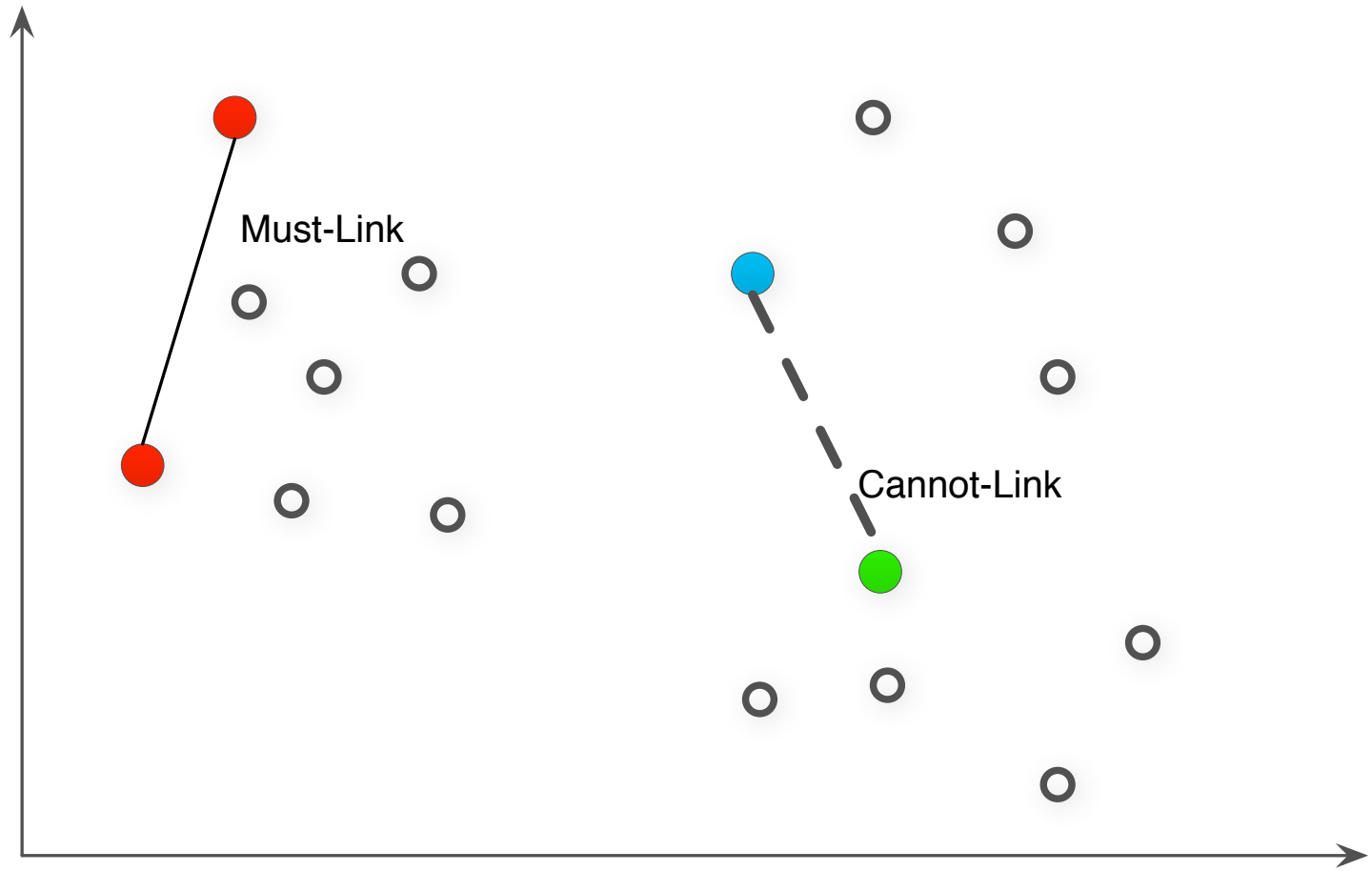
$\rightarrow$ Principalement des méthodes heuristiques dédiées à certains types de contraintes

Contraintes sur les clusters

- Contrainte de capacité: $\alpha \leq |C_i| \leq \beta$
- Contrainte du diamètre maximal
- Contrainte de marge minimale
- Contrainte de densité
- ...



Contraintes sur les paires de points



Fouille de données et méthodes déclaratives

- Recherche de motifs fréquents
  - 2008 : Premiers travaux de L. de Raedt sur la recherche d'itemsets fréquents en Programmation par Contraintes (PPC)
  - Extension au k-pattern-set mining *(Khiari &al., Guns & al.)*

  $\rightarrow$ Application au conceptual clustering

- Distance-based clustering (relationnel clustering)
  - 2 classes SAT *(Davidson & al)*
  - k classes avec k borné en PPC *(Dao, Duong & Vrain)*

Plan

Modélisation du clustering en 2-SAT

n données et 2 classes *(Davidson & al., 2010)*

- *n* variables booléennes x_i :
  $x_i = 0$ (resp. 1) si la donnée *i* est dans le cluster 0 (resp. 1)

- Contraintes
  - Contraintes $ML(x_i, x_j)$
    $(x_i \wedge x_j) \vee (\overline{x_i} \wedge \overline{x_j}) \rightarrow (x_i \vee \overline{x_j}) \wedge (\overline{x_i} \vee x_j)$
  - Contraintes $CL(x_i, x_j)$
    $(x_i \wedge \overline{x_j}) \vee (\overline{x_i} \wedge x_j) \rightarrow (x_i \vee x_j) \wedge (\overline{x_i} \vee \overline{x_j})$
  - Contraintes de diamètre $D \leq \alpha$:
    pour tout (i, j) tel que $d_{ij} > \alpha$, ajout de $CL(x_i, x_j)$
  - Contraintes de marge $S \geq \beta$:
    pour tout (i, j) tel que $d_{ij} < \beta$, ajout de $ML(x_i, x_j)$

Résolution

- Recherche des partitions satisfaisant les contraintes

- Minimisation du diamètre maximal :
  - Le diamètre maximal est l'une des dissimilarités.

  $\rightarrow$ Recherche dichotomique : appel du solveur avec la contrainte $D \leq x$
  - Complexité dans le pire des cas : $O(n^2 log(n))$

Autres approches

- Corrélation clustering: *(Berg et al. 2013)*
  $s_{ij} = 1$ si i et j sont similaires, 0 sinon.

→ Minimiser $\Sigma_{i,j|cl(i)=cl(j)}(1 - s_{ij}) + \Sigma_{i,j|cl(i)\neq cl(j)} s_{ij}$

→ Modéliser en Max-SAT

- $x_{ij} = 1$ si i et j sont dans le même cluster
- Contraintes dures pour exprimer $(x_{ij} = 1 \wedge x_{jk} = 1) \rightarrow x_{ik} = 1$
- Contraintes souples :
  - ★ x_{ij} pour chaque paire d'objets vérifiant $s_{ij} = 1$
  - ★ $\overline{x_{ij}}$ pour chaque paire d'objets vérifiant $s_{ij} = 0$

- Un langage à base de contraintes pour divers problèmes de Data Mining *(Méthivier & al., 2012)*

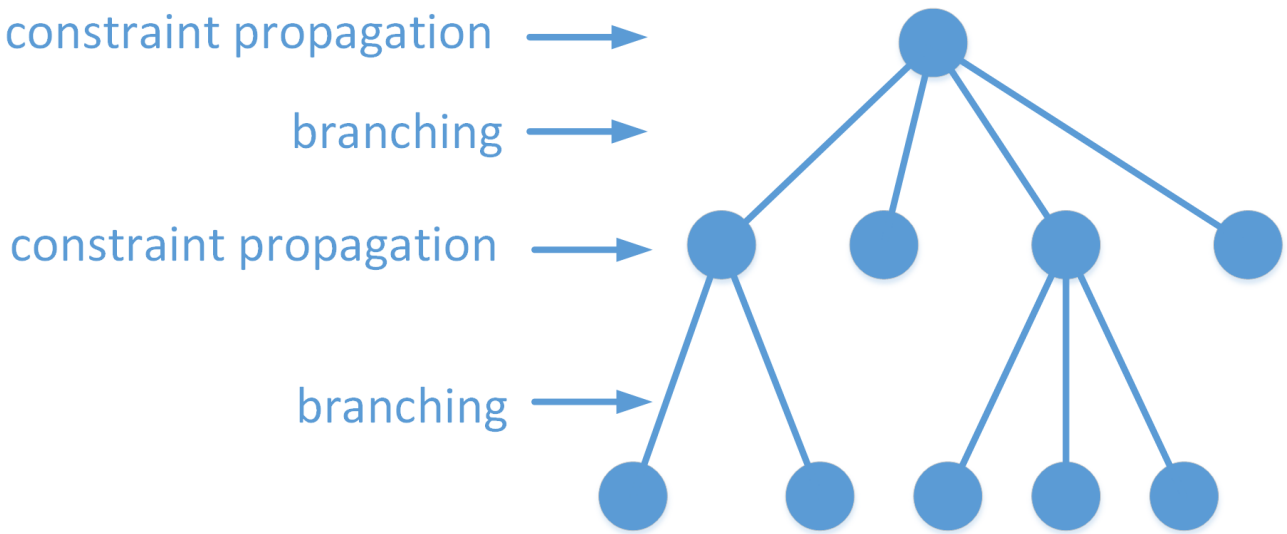
→ application au clustering conceptuel

- Modélisation en PPC puis en SAT

Plan

Programmation par Contraintes (PPC)

- Modélisation déclarative du problème en spécifiant **variables** et **contraintes**
- Recherche des solutions par **propagation de contraintes** et **branchement**



PPC: Exemple

Deux types de problème :

- Trouver toutes les affectations de chiffres aux lettres de sorte que

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

$\rightarrow$ CSP : problème de satisfaction de contraintes

- Trouver une affectation de chiffres aux lettres de sorte que
  - $SEND + MOST = MONEY$
  - $MONEY$ est maximisé.

$\rightarrow$ COP : problème d'optimisation sous contraintes

PPC : Modélisation de l'exemple

- Variables $S, E, N, D, M, O, T, Y \in \{0, \ldots, 9\}$
- Variable V pour représenter la fonction objectif
- Contraintes:
  - $S \neq 0,\ M \neq 0$
  - $\texttt{alldifferent}(S, E, N, D, M, O, T, Y)$
  - contrainte linéaire

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

  - $V = 10000 \times M + 1000 \times 0 + 100 \times N + 10 \times E + Y$
- Fonction objectif: *maximiser V*

PPC : solveurs

- Les solveurs de PPC trouvent des solutions en itérant
  - propagation de contraintes : suppression de valeurs inconsistantes de domaines de variables

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

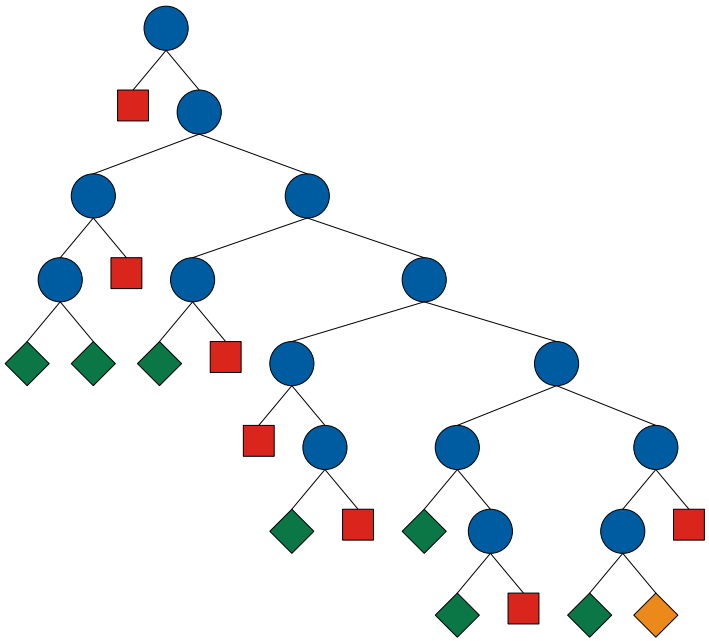
  - branchement: création de branches dans l'arbre de recherche

$$D_E = \{2\} \quad \text{and} \quad D_E = \{3,4,5,6,7\}$$

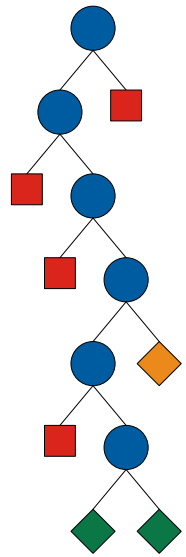
- Les stratégies de branchement peuvent être définies par l'utilisateur.

Influence des stratégies de recherche

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



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



état stable
état échec
solution intermédiaire
meilleure solution

Contraintes Globales

Contraintes globales

Contraintes encapsulant un ensemble de contraintes
$\Rightarrow$ des algorithmes de filtrage plus puissants

Exemple: Les valeurs sont 2 à 2 distinctes

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

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

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

Clustering conceptuel

$\mathcal{T}$: ens. de n objets décrits par m propriétés booléennes ($\mathcal{I}$)

	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

- motif : ens. de propriétés $p_1, \ldots, p_j$, fermé si tous les objets satisfaisant $p_1, \ldots p_j$ n'ont que ces propriétés en commun.
- concept : (ens. d'objets, motif fermé) tel que les objets, et seulement ces objets, vérifient le motif

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

$\rightarrow$ Recherche de k motifs fermés (induisant k concepts) qui satisfont un ensemble de contraintes

Clustering conceptuel : modélisation

$\mathcal{T}$: ensemble de n objets (transactions)

Recherche de k motifs $\Pi = (\pi_1, \ldots, \pi_k)$

- tels que
  - $\forall \pi_i$, *couverture*(π_i)
  - $\forall \pi_i$, *fermé*(π_i)
  - *couverture*$(\Pi) = \mathcal{T}$
  - $\forall \pi_i$, $\forall \pi_i$ avec $i \neq j$, *overlap*$(\pi_i, \pi_j) = 0$

- optimisant
  - maximisant $min(freq(\pi_1), \ldots, freq(\pi_k))$ ou
  - maximisant
    $max(freq(\pi_1), \ldots, freq(\pi_k)) - min(freq(\pi_1), \ldots, freq(\pi_k))$

Clustering conceptuel : modélisation

→ Variables

- $k \times n$ variables binaires T_i^p : la donnée i appartient au cluster p
- $k \times m$ variables binaires I_j^p : la propriété j décrit le cluster p

→ Contraintes

- Contraintes de couverture
  $\forall p \in [1..k], \forall t \in \mathcal{T}, T_t^p \leftrightarrow \Sigma_{i \in \mathcal{I}} I_i^p (1 - D_{t,i}) = 0$
- Contraintes de fermeture
  $\forall p \in [1..k], \forall i \in \mathcal{I}, I_i^p \leftrightarrow \Sigma_{t \in \mathcal{T}} T_t^p (1 - D_{t,i}) = 0$
- Les motifs doivent couvrir tous les objets.
- Contrainte sur la taille des motifs : $\forall p \in [1, k], \Sigma_{i \in \mathcal{I}} I_i^p \leq \alpha$
- Contrainte sur la taille des clusters : $\forall p \in [1, k], \Sigma_{t \in \mathcal{T}} T_t^p \leq \alpha$
- Partition, Modèle canonique . . .

- Expérimentations :
  - primary-tumor (336, 31, $k \in [1, 4]$), audiology (216, 148, $k \in [1, 7/4]$), anneal (812, 53, $k \in [1, 7/6]$)
  - 2 critères d'optimisation : *min taille*, *écart taille*
  - *min taille* est plus efficace que *écart taille*

Clustering relationnel en PPC

Entrées :

- une mesure de dissimilarité entre paires d'objets
- une borne sur le nombre de classes : $k_{min} \leq k \leq k_{max}$

Sortie : partition des données

- intégration de divers types de contraintes
- choix d'un critère d'optimisation parmi :
  - Minimiser le diamètre maximal des clusters
  - Maximiser la marge minimale entre clusters
  - Minimiser la somme des dissimilarités intra-cluster

Application : clustering bi-critère marge-diamètre ($\max S, \min D$)

Variables

- n variables entières : $\mathcal{G} = [G_1, \ldots, G_n]$, $dom(G_i) = \{1, \ldots, k_{max}\}$.
  $G_i = c$: le point i est affecté à la c-ème classe.

- Une variable pour le critère à optimiser:
  - D : diamètre maximal
  - S : marge minimale
  - V : somme des dissimilarités intra-cluster
  - $dom(D) = dom(S) = [\min_{i,j}(d(i,j)), \max_{i,j}(d(i,j))]$
  - $dom(V) = [0, \sum_{i<j} d(i,j)]$.

Contraintes de partitionnement

- Casser les symétries de valeurs :

Contrainte *precede* : $G_1 = 1$ et $G_i \leq \max_{j \in [1,i-1]}(G_j) + 1$, pour $i \in [2, n]$

- $precede(\mathcal{G}, [1, \ldots, k_{max}])$
  signifie $G_1 = 1$ et $\forall i \in [2, n]$, si $G_i = c$ alors $existsj < i$ $G_j = c - 1$

- Au moins k_{min} clusters :

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

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

Contraintes utilisateur

Taille minimale α des clusters : $\forall i \in [1, n], \#\{j \mid G_j = G_i\} \geq \alpha$

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

Diamètre maximal γ des clusters :

- $D \leq \gamma$
- pour tout i, j tel que $d(i, j) > \gamma$, on pose la contrainte $G_i \neq G_j$

Contraintes sur les couples de points

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

Critère d'optimisation

- Stratégie *branch-and-bound* pour optimiser un critère
  nouvelle solution $\rightarrow$ ajout de contraintes pour interdire des solutions moins bonnes.

Solution Δ trouvée

- calcul de $D(\Delta)$
- ajout de la contrainte $D < D(\Delta)$
- pour chaque couple (i, j), $i < j \in [1, n]$ tel que $d(i, j) \geq D(\Delta)$
  ajout de la contrainte : $G_i \neq G_j$.

Stratégie de recherche

- ordonnancement initial des points $\rightarrow$ utilisation de FPF

- stratégies de choix des variables
  - optimisation de WCSD :
    - ★ recherche gloutonne pour trouver rapidement une "bonne" 1ère solution
    - ★ puis détection rapide des échecs

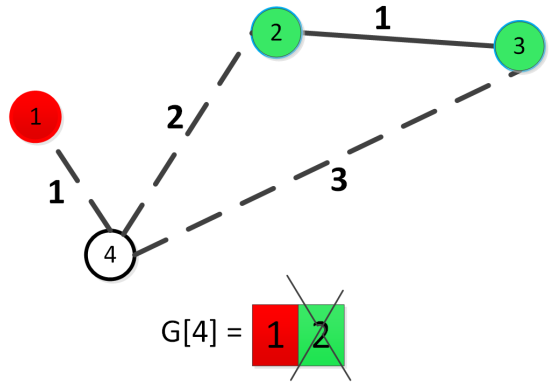
- développement d'algorithmes de filtrage
  - diamètre : $\forall i < j \in [1, n], \quad D < d(i,j) \rightarrow G_i \neq G_j$
    - ★ implantation par des contraintes réifiées
    - $\rightarrow$ un algorithme de filtrage dédié
  - WCSD : $V = \Sigma_{i<j}(G_i == G_j)d(i,j)$
  - $\rightarrow$ développement d'un algorithme pour améliorer le filtrage

Propagation insuffisante

- Supposons qu'on a trouvé une solution avec $V = 5$, la stratégie branch-and-bound ajoute une nouvelle contrainte

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

  - puisque les points 2 et 3 sont dans le même cluster, $(G_2 == G_4) = (G_3 == G_4)$
  - et donc le point 4 ne peut être dans le cluster 2



- Or la valeur 2 n'est pas supprimée du domaine de G_4

Expérimentations

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

Expérimentations

- Minimisation du diamètre maximal
- Comparaison entre
  - BaB : approche *branch-and-bound (Brusco 2005)*
  - GC : coloriage de graphe *(Hansen 1980)*
  - CP1 : premier modèle *(Dao, Duong & Vrain, ECML 2013, ICTAI2013)*
  - CP2 : second modèle *(Dao, Duong & Vrain, RIA 2014)*

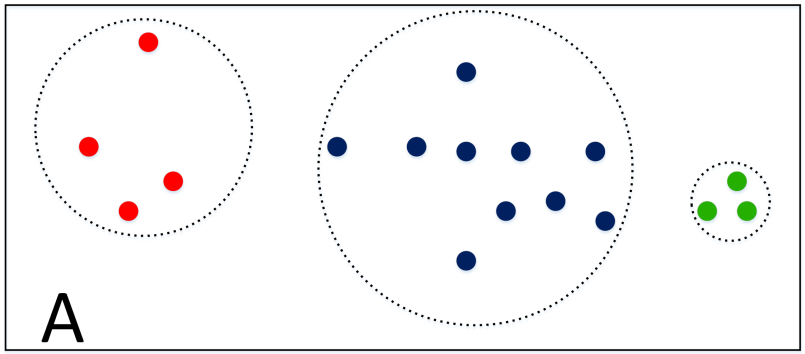
Expérimentations

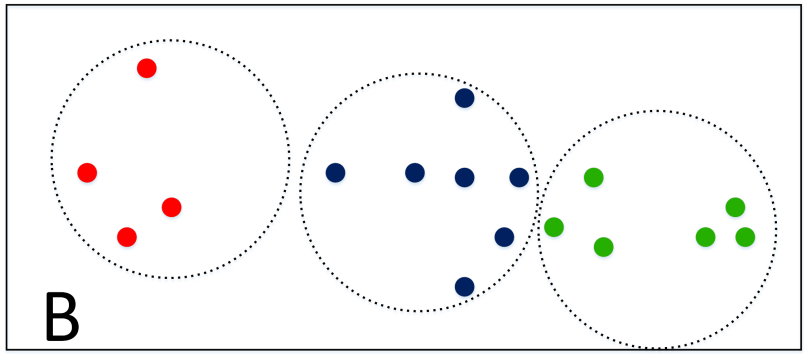
Dataset	D_{opt}	BaB	GC	CP1	CP2
Iris	2.58	1.4	1.8	< 0.1	< 0.1
Wine	458.13	2	2.3	0.3	< 0.1
Glass	4.97	8.1	42	0.9	0.2
IonoSphere	8.6	–	0.6	0.4[1]	0.3
User Knowledge	1.17	–	3.7	75	0.2
Breast Cancer	2377.96	–	1.8	0.7	0.5
Synthetic Control	109.36	–	–	56.1	1.6
Vehicle	264.83	–	–	14.3	0.9
Yeast	0.67	–	–	2389.9	5.2
Multi Features	12505.5	–	–	$*$	10.4
Image Segmentation	436.4	–	–	589.2	5.7
Waveform	15.6	–	–	$*$	50.1

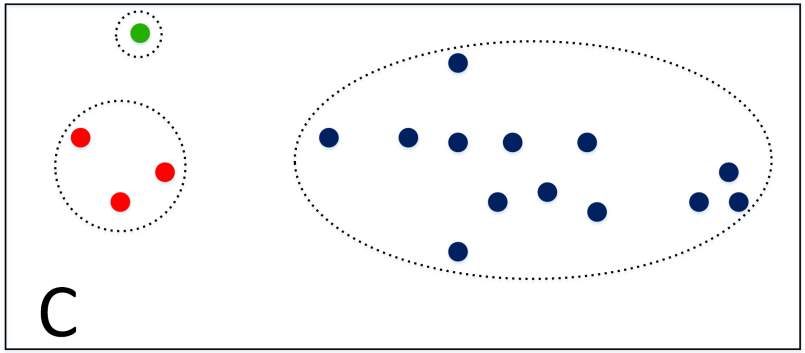
Table: Performance (en secondes) - minimisation du diamètre maximal

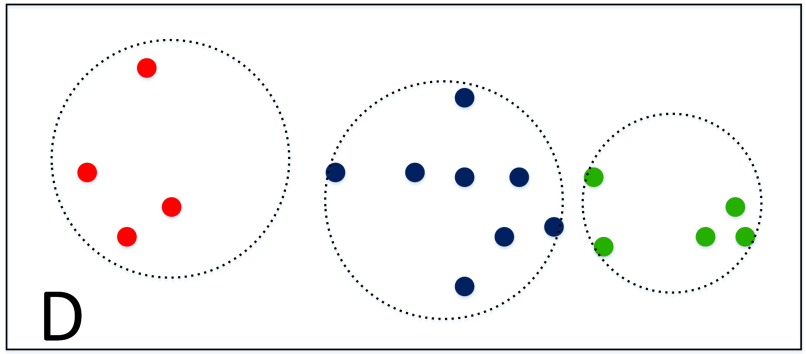
Classification non supervisée bi-critère

Effets indésirables des différents critères









Clustering bi-critère marge-diamètre

Clustering bi-critère marge-diamètre : $(\min D, \max S)$

$\rightarrow$ chercher les solutions de Pareto

- Une partition Δ' domine une partition Δ si et seulement si:
  $D(\Delta') \leq D(\Delta)$ et $S(\Delta') > S(\Delta)$ ou
  $D(\Delta') < D(\Delta)$ et $S(\Delta') \geq S(\Delta)$.
- Δ est une solution optimale de Pareto ssi il n'existe pas de partition Δ' qui domine Δ
- front de Pareto = $\{(D(P), S(P)) \mid \text{P solution optimale Pareto}\}$

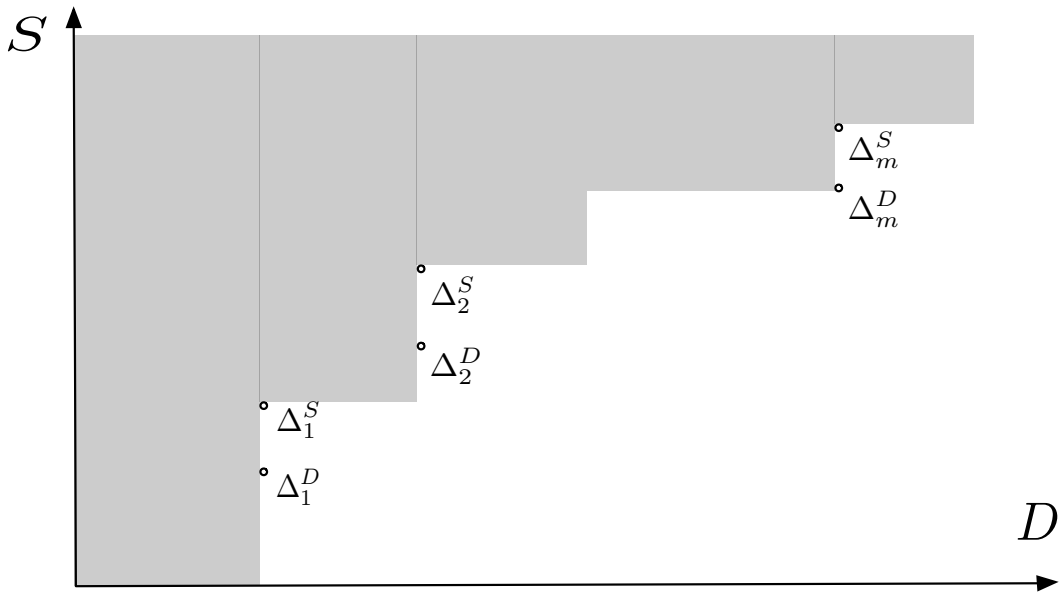
Bi-critère de la marge et du diamètre

- clustering bi-critère sous contraintes utilisateur $\mathcal{C}$
- itération du modèle en ajoutant des contraintes

$\rightarrow$ intérêt d'un cadre déclaratif

Algorithme

$\mathcal{A} \leftarrow \emptyset$;
$i \leftarrow 1$;
$\Delta_i^D \leftarrow$ 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($\mathcal{C} \cup \{S > S(\Delta_{i-1}^S)\}$);



Expérimentation

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: meilleur algorithme exact connu (Delattre *et al.*, 1980)
- Temps en secondes, time out 1 heure

Et le critère WCSS ?

- RBBA : Repetitive Branch and Bound Algorithm *(Brusco, 2006)*, sans contraintes utiisateur
  - Réordonnancement des objets : le plus près à des extrémités opposés dans l'ordonnancement
  - Itérativement, résoudre le problème pour $k+1, \ldots, k+n$ objets.
    - ★ La valeur optimale obtenue à une étape donne une borne pour l'étape suivante.
    - ★ A chaque étape, algorithme Branch & Bound pour trouver une solution optimale $WCSS^*(T_1 \cup T_2) \geq WCSS^*(T_1) + WCSS^*(T_2)$
  - Expérimentations :
    - ★ clusters bien séparés : n=240, k=8
    - ★ pas de structure sous-jacente : n=60, k=6
- Integer Linear Programming avec contraintes utilisateur : *(Babaki & al. 2014)*
  - Introduction de contraintes monotones et anti-monotones
  - Expérimentations sur iris, wine, soybean + contraintes

Conclusion

- Intérêt des modèles déclaratifs pour la modélisation de contraintes utilisateurs

$\rightarrow$ Interaction avec les utilisateurs

$\rightarrow$ Résolution de problèmes bi-critères

- Importance de la modélisation, des stratégies de recherche et de filtrage
- Des critères d'optimisations plus faciles que d'autres.
- Taille des bases de données ?
  - "smart data", "valuable data" : données difficiles à obtenir et donc rares
  - big data ?
    - ★ parallélisation des solveurs
    - ★ recherche heuristique dans les solveurs