Federico Brandini (mat. 358443)
Alessandro Mazzocchi (mat. 367081)

Raggruppamento di Punti

i. Testo del problema

Sia data una nuvola di punti nello spazio con distanza d_{ij} tra ogni coppia distinta di punti. Vogliamo partizionare i punti in K sottoinisiemi in modo tale da:

1. minimizzare la massima somma delle distanze tra punti nello stesso sottoinsieme;
2. massimizzare la minima distanza tra punti in sottoinsiemi diversi.

1. Modello matematico

Chiamiamo P l'insieme degli indici riferiti ai diversi punti. I dati del problema sono le distanze $d_{ij} > 0$ (positive in quanto non ci possono essere più occorrenze dello stesso punto) per ogni coppia di punti $(i, j) \in P^2$ con $i < j$ (data la simmetria tra le distanze nello spazio euclideo) e il numero naturale di raggruppamenti $K \geq 2$ (il caso $K = 0$ non ha alcun senso e il caso $K = 1$ è banale). Inoltre, imponiamo anche che $K \leq |P|$ per evitare la formazione di raggruppamenti certamente vuoti. La condizione finale su K sarà dunque

$$2 \leq K \leq |P|;$$

è dunque inutile imporre che ci siano almeno due punti in P. Per semplicità, identifichiamo il k-esimo raggruppamento con P_k; si avrà dunque che

$$\bigcup_{k=1}^{K} P_k = P, \qquad P_{k_1} \cap P_{k_2} = \emptyset \ \forall k_1, k_2 = 1, \dots, K, k_1 \neq k_2,$$

cioè, che i vari raggruppamenti costituiscono una partizione di P.

Associamo ad ogni coppia punto-raggruppamento (i, k) la variabile binaria

$$x_{ik} = \begin{cases} 1, & i \in P_k \\ 0, & i \notin P_k \end{cases}.$$

Imponiamo, quindi, che

- ogni raggruppamento debba contenere almeno un punto, cioè che $|P_k| \geq 1$:

$$\sum_{i \in P} x_{ik} \geq 1 \quad \forall k = 1, \dots, K;$$

- ogni punto debba appartenere a uno e un solo raggruppamento:

$$\sum_{k=1}^{K} x_{ik} = 1 \quad \forall i \in P.$$

Ora ci occupiamo dell'obiettivo da massimizzare (per nostra scelta) e, siccome questo sarà la somma pesata di due sotto-obiettivi, procediamo ad una formulazione separata di questi ultimi.

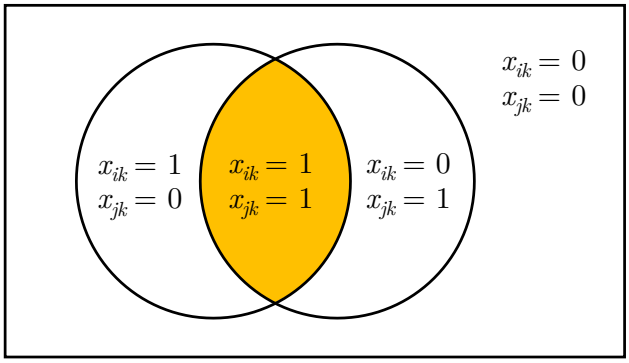
1. Il primo è da minimizzare ed è la massima somma delle distanze tra punti nello stesso sottoinsieme:

$$\max_{k=1,\dots,K} \sum_{\substack{i,j\in P_k \\ i<j}} d_{ij}\,.$$

Consideriamo l'argomento della funzione max e cerchiamo di renderlo lineare.

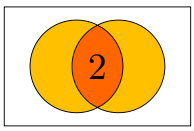
$$\sum_{\substack{i,j\in P_k \\ i<j}} d_{ij} = \sum_{\substack{i,j\in P \\ i<j \\ x_{ik}=x_{jk}=1}} d_{ij} = (*)$$

Per rendere più semplice il compito, rappresentiamo l'insieme delle distanze che dovranno essere sommate in questo modo.



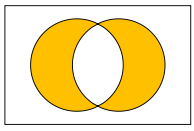
Inizialmente, potremmo sommare le distanze d_{ij} per le quali $x_{ik}=1$ con quelle per cui $x_{jk}=1$; indicando con B il valore di tale somma, si ha che

$$B = \sum_{\substack{i,j\in P \\ i<j \\ x_{ik}=1}} d_{ij} + \sum_{\substack{i,j\in P \\ i<j \\ x_{jk}=1}} d_{ij} = \sum_{\substack{i,j\in P \\ i<j}} d_{ij}x_{ik} + \sum_{\substack{i,j\in P \\ i<j}} d_{ij}x_{jk}$$
$$= \sum_{\substack{i,j\in P \\ i<j}} d_{ij}(x_{ik}+x_{jk}).$$

B 

In secondo luogo, togliamo dalla somma di tutte le distanze la somma di quelle per cui $x_{ik}=x_{jk}$; indicando con A tale differenza, si ha che

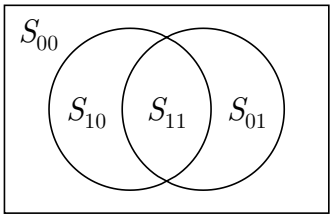
$$A = \sum_{\substack{i,j\in P \\ i<j}} d_{ij} - \sum_{\substack{i,j\in P \\ i<j \\ x_{ik}=x_{jk}}} d_{ij}.$$

A 

Il valore cercato è, quindi,

$$(*) = \frac{1}{2}(B-A).$$

Ora, il problema di rendere lineare l'espressione $(*)$ si risolve rendendo lineare il sottraendo dell'espressione di A: quello indicato in rosso. Siano $S_{00}, S_{10}, S_{11}, S_{01}$ le



somme delle distanze in ognuna delle loro quattro partizioni; allora

$$\sum_{\substack{i,j\in P\\ i<j\\ x_{ik}=x_{jk}}} d_{ij} = S_{00} + S_{11}$$

e, siccome attraverso espressioni lineari è possibile coprire i casi base $S_{00}+S_{10}, S_{00}+S_{01}, S_{10}+S_{11}$ e le relative combinazioni lineari soltanto, dobbiamo verificare se la sommatoria rossa rientra fra queste. Per farlo, utilizziamo l'applicazione lineare delle coordinate rispetto alla base canonica definita come segue:

$$[aS_{00} + bS_{10} + cS_{11} + dS_{01}]_{\mathcal{C}} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}.$$

Risolviamo dunque il seguente sistema:

$$[[S_{00}+S_{10}]_{\mathcal{C}} \quad [S_{00}+S_{01}]_{\mathcal{C}} \quad [S_{10}+S_{11}]_{\mathcal{C}}]x = [S_{00}+S_{11}]_{\mathcal{C}}.$$

Il sistema diventa

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} x = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

che è impossibile. Di conseguenza, è necessario introdurre una non linearità da gestire successivamente, per esempio, in questo modo:

$$\sum_{\substack{i,j\in P\\ i<j\\ x_{ik}=x_{jk}}} d_{ij} = \sum_{\substack{i,j\in P\\ i<j}} d_{ij}(1-|x_{ik}-x_{jk}|) = \sum_{\substack{i,j\in P\\ i<j}} d_{ij}(1-y_{ijk})\,.$$

Per garantire che $y_{ijk} = |x_{ik} - x_{jk}|$ è sufficiente imporre i seguenti quattro nuovi vincoli (tutti necessari):

$$\begin{aligned} &y_{ijk} \geq x_{ik} - x_{jk} \\ &y_{ijk} \geq -x_{ik} + x_{jk} \\ &y_{ijk} \leq 2 - x_{ik} - x_{jk} \\ &y_{ijk} \leq 2 - (1 - x_{ik}) - (1 - x_{jk}), \qquad \forall i,j \in P : i < j, \forall k = 1, \ldots, K. \end{aligned}$$

Verifichiamolo qui di seguito (notare che le ultime due colonne coincidono).

x_{ik}	x_{jk}	vincoli	y_{ijk}	$\lvert x_{ik} - x_{jk} \rvert$
0	0	$\boldsymbol{y_{ijk} \geq 0}$ $\boldsymbol{y_{ijk} \geq 0}$ $y_{ijk} \leq 2$ $\boldsymbol{y_{ijk} \leq 0}$	0	0
0	1	$y_{ijk} \geq -1$ $\boldsymbol{y_{ijk} \geq 1}$ $\boldsymbol{y_{ijk} \leq 1}$ $\boldsymbol{y_{ijk} \leq 1}$	1	1
1	0	$\boldsymbol{y_{ijk} \geq 1}$ $y_{ijk} \geq -1$ $\boldsymbol{y_{ijk} \leq 1}$ $\boldsymbol{y_{ijk} \leq 1}$	1	1
1	1	$\boldsymbol{y_{ijk} \geq 0}$ $\boldsymbol{y_{ijk} \geq 0}$ $\boldsymbol{y_{ijk} \leq 0}$ $y_{ijk} \leq 2$	0	0

Ora l'espressione di A è lineare e può essere riscritta sottoforma di un'unica sommatoria:

$$A = \sum_{\substack{i,j \in P \\ i<j}} d_{ij} - \sum_{\substack{i,j \in P \\ i<j \\ x_{ik}=x_{jk}}} d_{ij} = \sum_{\substack{i,j \in P \\ i<j}} d_{ij} - \sum_{\substack{i,j \in P \\ i<j}} d_{ij}(1 - y_{ijk}) = \sum_{\substack{i,j \in P \\ i<j}} d_{ij} y_{ijk} \,.$$

Riscriviamo, quindi, l'espressione dell'argomento della funzione max nell'obiettivo in forma lineare:

$$(*) = \frac{1}{2}(B - A) = \frac{1}{2} \sum_{\substack{i,j \in P \\ i<j}} d_{ij}(x_{ik} + x_{jk} - y_{ijk}) \,.$$

L'obiettivo 1 diventa

$$z_1 = \max_{k=1,\dots,K} \frac{1}{2} \sum_{\substack{i,j \in P \\ i<j}} d_{ij}(x_{ik} + x_{jk} - y_{ijk}) \,;$$

per linearizzarlo procediamo nel seguente modo:

$$\begin{aligned} &\min \quad z_1 \\ &z_1 \geq \frac{1}{2} \sum_{\substack{i,j \in P \\ i<j}} d_{ij}(x_{ik} + x_{jk} - y_{ijk}) \,, \qquad \forall i, j \in P : i < j, \forall k = 1, \dots, K. \end{aligned}$$

Infine, siccome il problema generale è un problema di massimo, compiamo l'ultima azione:

$$\min \quad z_1 = -\max \ -z_1.$$

2. Il secondo è da massimizzare ed è la minima distanza tra punti in sottoinsiemi diversi:

$$\min_{\substack{k=1,\dots,K \\ i,j\in P: i<j \\ i\in P_k \\ j\notin P_k}} d_{ij} = \min_{\substack{k=1,\dots,K \\ i,j\in P: i<j}} \frac{d_{ij}}{2}\Big(x_{ik} + (1 - x_{jk})\Big) + M(1 - x_{ik}) + Mx_{jk}\,,$$

dove $M = \max_{\substack{i,j\in P \\ i<j}} d_{ij}$. L'idea generale è stata quella di "escludere" le distanze d_{ij} tali che $\neg(i \in P_k \wedge j \notin P_k)$ aggiungendo loro (o sostituendole con) un valore sufficientemente grande, ossia un'espressione uguale o superiore a M. Verifichiamo caso per caso considerando l'argomento della funzione min nell'obiettivo

$$\frac{d_{ij}}{2}\Big(x_{ik} + (1 - x_{jk})\Big) + M(1 - x_{ik}) + Mx_{jk} = (**):$$

- $x_{ik} = x_{jk} = 0 \Rightarrow (**) = \frac{d_{ij}}{2} + M \geq M$ (in realtà è anche strettamente maggiore, dato che $d_{ij} > 0$);
- $x_{ik} = 0 \wedge x_{jk} = 1 \Rightarrow (**) = 2M \geq M$ (in realtà è anche strettamente maggiore, dato che $M > 0$);
- $x_{ik} = 1 \wedge x_{jk} = 0 \Rightarrow (**) = d_{ij}$;
- $x_{ik} = x_{jk} = 1 \Rightarrow (**) = \frac{d_{ij}}{2} + M \geq M$ (in realtà è anche strettamente maggiore per lo stesso motivo del primo caso).

L'obiettivo 2 è, quindi,

$$z_2 = \min_{\substack{k=1,\dots,K \\ i,j\in P: i<j}} \frac{d_{ij}}{2}\Big(x_{ik} + (1 - x_{jk})\Big) + M(1 - x_{ik}) + Mx_{jk}\,;$$

per linearizzarlo procediamo nel seguente modo:

$$\begin{aligned}
&\max \quad z_2 \\
&z_2 \leq \frac{d_{ij}}{2}\Big(x_{ik} + (1 - x_{jk})\Big) + M(1 - x_{ik}) + Mx_{jk}, \qquad \forall i,j \in P : i < j, \forall k = 1,\dots,K.
\end{aligned}$$

Siccome il problema generale è di massimo non bisogna compiere alcuna azione ulteriore, diversamente da quanto fatto per l'obiettivo 1.

Ora possiamo scrivere il modello matematico completo per questo problema formulando l'obiettivo complessivo come una somma pesata dei due ($c_1, c_2 \geq 0$ non entrambi nulli).

$$\begin{aligned}
&\max \quad c_1(-z_1) + c_2 z_2 \\
&\sum_{i\in P} x_{ik} \geq 1 \quad \forall k = 1,\dots,K \\
&\sum_{k=1}^{K} x_{ik} = 1 \quad \forall i \in P
\end{aligned}$$

$$
\begin{aligned}
&y_{ijk} \geq x_{ik} - x_{jk} \\
&y_{ijk} \geq -x_{ik} + x_{jk} \\
&y_{ijk} \leq 2 - x_{ik} - x_{jk} \\
&y_{ijk} \leq 2 - (1 - x_{ik}) - (1 - x_{jk}) \quad \forall i, j \in P : i < j, \forall k = 1, \dots, K
\end{aligned}
$$

$$
z_1 \geq \frac{1}{2} \sum_{\substack{i,j \in P \\ i<j}} d_{ij}(x_{ik} + x_{jk} - y_{ijk}) \quad \forall i, j \in P : i < j, \forall k = 1, \dots, K
$$

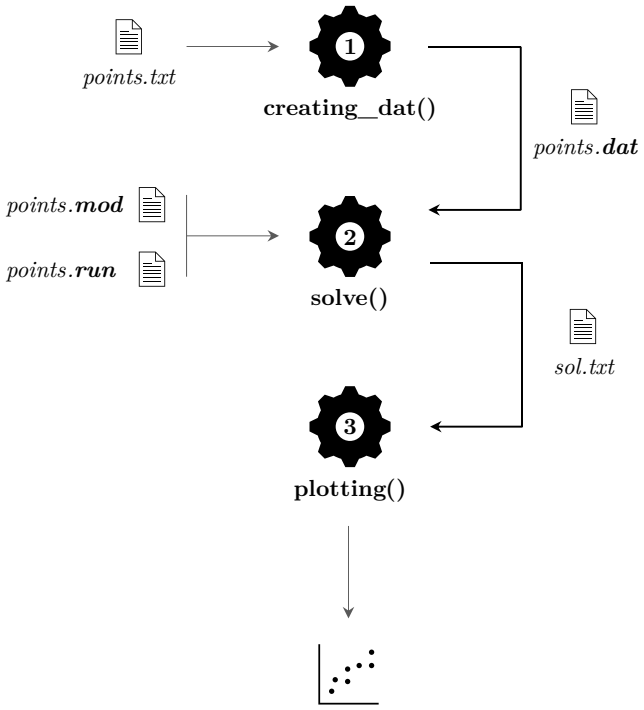
$$
z_2 \leq \frac{d_{ij}}{2}\Big(x_{ik} + (1 - x_{jk})\Big) + M(1 - x_{ik}) + M x_{jk} \quad \forall i, j \in P : i < j, \forall k = 1, \dots, K
$$

$$
\begin{aligned}
&x_{ik} \in \{0,1\} \quad \forall i \in P, \forall k = 1, \dots, K \\
&y_{ijk} \in \{0,1\} \quad \forall i, j \in P : i < j, \forall k = 1, \dots, K \\
&z_1, z_2 \in \mathbb{R}
\end{aligned}
$$

Si tratta di un problema di Programmazione Lineare Mista Intera (PLMI).

2. Software ausiliario

Per visualizzare graficamente la soluzione del problema, abbiamo creato un semplice script Python *encapsulator.py* per eseguire il plotting dei punti su un grafico con colori diversi a seconda dell'insieme a cui appartengono. Il programma svolge le seguenti tre funzioni principali.



1. **Creazione del file *points.dat*** dalla lettura di un file *points.txt* contenente i punti avente il seguente formato ($m = |P|$);

$$
\begin{array}{|l|}
\hline
K \\
n \\
coo_{11}; coo_{12}; \dots; coo_{1n} \\
coo_{21}; coo_{22}; \dots; coo_{2n} \\
\vdots \\
coo_{m1}; coo_{m2}; \dots; coo_{mn} \\
\hline
\end{array}
$$

2. **risoluzione del problema** con stampa della soluzione (solo variabili x_{ik}) in un file *sol.txt*;

3. **visualizzazione grafica** della soluzione su un grafico n-dimensionale con n uguale a 2 o 3.

3. Risoluzione di un'istanza

Consideriamo il seguente insieme di punti di $\mathbb{R}^3$:

$$\{(2, -1, 6.5), (-1.5, 0, 2), (7.5, 2, 0), (6, 7, 1), (5, 4.5, 1), (2, 8, 8)\};$$

partiamo da questo per definire l'istanza del nostro problema, che è rappresentata nella seguente tabella di distanze (qui riportate approssimate alla seconda cifra decimale) e dal numero di partizioni $K = 3$. Fissiamo inizialmente i pesi c_1 e c_2 a 1.

d_{ij}	1	2	3	4	5	6
1		5.79	9.03	10.5	8.34	9.12
2			9.43	10.31	7.97	10.59
3				5.31	3.67	11.41
4					2.69	8.12
5						8.38
6						

Impostiamo opportunamente il file *points.txt* e avviamo lo script per la risoluzione dell'istanza (il risolutore scelto è Gurobi). Osservando l'output nel file *sol.txt*, le tre partizioni individuate nella soluzione ottima restituita sono $\{(\mathbf{7.5}, \mathbf{2}, \mathbf{0}), (\mathbf{6}, \mathbf{7}, \mathbf{1}), (\mathbf{5}, \mathbf{4.5}, \mathbf{1})\}$, $\{(\mathbf{2}, \mathbf{-1}, \mathbf{6.5}), (\mathbf{-1.5}, \mathbf{0}, \mathbf{2})\}$ e $\{(\mathbf{2}, \mathbf{8}, \mathbf{8})\}$.

Vediamo ora come cambia la soluzione ottima del problema al variare dei pesi attribuiti ai due sotto-obiettivi. In questa fase considereremo solo cambiamenti significativi della soluzione ottima, nel senso che non identificheremo i cambiamenti nell'ennupla ottima che rimandano agli

stessi raggruppamenti; per esempio, le seguenti due soluzioni ottime sono in realtà la stessa (l'unica cosa che cambia è l'ordine delle partizioni).

$$\{(\mathbf{7.5, 2, 0}), (\mathbf{6, 7, 1}), (\mathbf{5, 4.5, 1})\}, \{(\mathbf{2, -1, 6.5}), (\mathbf{-1.5, 0, 2})\}, \{(\mathbf{2, 8, 8})\}$$
$$\{(\mathbf{2, -1, 6.5}), (\mathbf{-1.5, 0, 2})\}, \{(\mathbf{2, 8, 8})\}, \{(\mathbf{7.5, 2, 0}), (\mathbf{6, 7, 1}), (\mathbf{5, 4.5, 1})\}$$

Per semplicità, facciamo dipendere i pesi c_1 e c_2 dei due sotto-obiettivi da un unico parametro c; a tal proposito consideriamo la funzione obiettivo $c_1(-z_1) + c_2 z_2$ e dividiamola per $c_1 + c_2$, ottenendo così

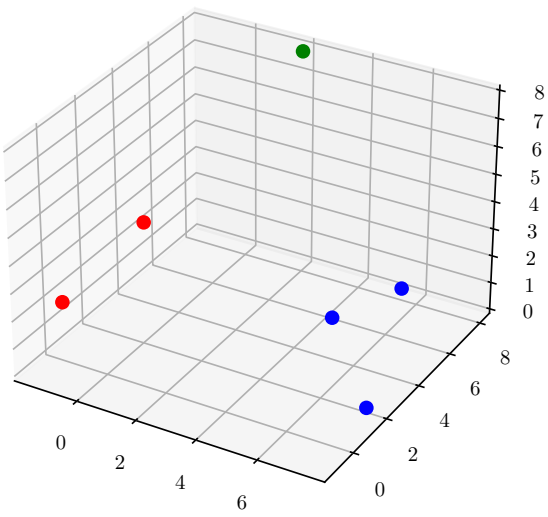
$$\frac{c_1}{c_1 + c_2}(-z_1) + \frac{c_2}{c_1 + c_2} z_2.$$

Ponendo $c = \frac{c_1}{c_1 + c_2}$, possiamo riscrivere l'obiettivo in questo modo:

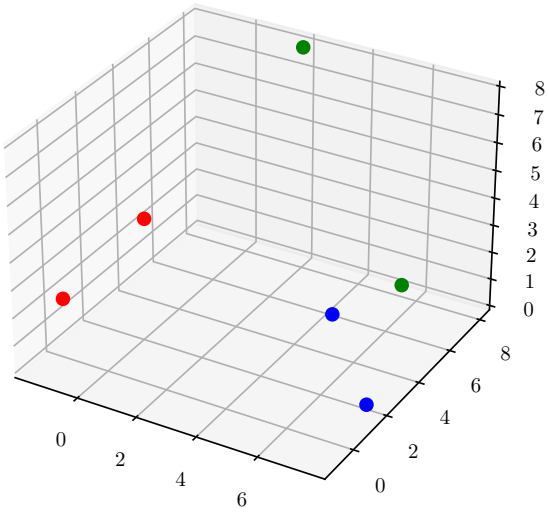
$$c(-z_1) + (1 - c)z_2, \qquad \text{con } c \in [0, 1].$$

Al variare di c otteniamo due diversi raggruppamenti ottimi:

- $\{(\mathbf{7.5, 2, 0}), (\mathbf{6, 7, 1}), (\mathbf{5, 4.5, 1})\}, \{(\mathbf{2, -1, 6.5}), (\mathbf{-1.5, 0, 2})\}, \{(\mathbf{2, 8, 8})\}$ per $0 \leq c \leq 0.59725$;
- $\{(\mathbf{7.5, 2, 0}), (\mathbf{5, 4.5, 1})\}, \{(\mathbf{2, -1, 6.5}), (\mathbf{-1.5, 0, 2})\}, \{(\mathbf{6, 7, 1}), (\mathbf{2, 8, 8})\}$ per $0.59725 < c \leq 1$.



$0 \leq c \leq 0.59725$



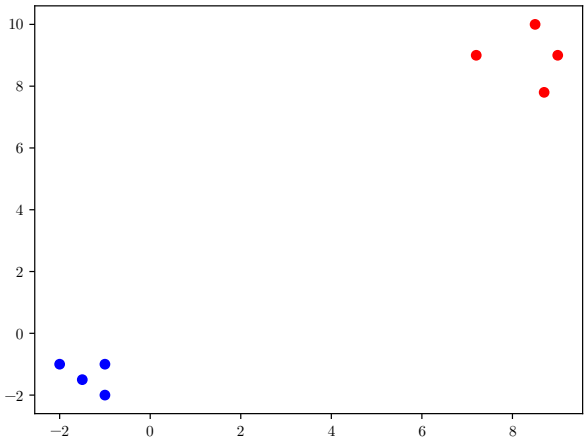
$0.59725 < c \leq 1$

4. Una considerazione generale

Cerchiamo inizialmente di dare una descrizione intuitiva dei due sotto-obiettivi:

1. il primo privilegia le soluzioni nelle quali le partizioni hanno alta densità, cioè quelle dove i punti dello stesso raggruppamento sono più vicini tra loro;
2. il secondo privilegia le soluzioni nelle quali le partizioni sono tra loro più distanti.

Dunque, regolando opportunamente i pesi dei due obiettivi, è possibile ottenere una soluzione ottima piuttosto che un'altra, a seconda delle nostre esigenze e del caso in cui ci troviamo; tuttavia, sembrerebbe esistere un caso in cui i raggruppamenti non cambiano al variare dei pesi c_1 e c_2,



Caso bidimensionale con $K = 2$

dove si riconoscono K addensamenti di punti tali che i punti in ogni addensamento siano sufficientemente vicini e/o che i diversi addensamenti siano sufficientemente lontani fra loro. Ovviamente questa considerazione è basata unicamente sulle sperimentazioni.

5. Complessità del modello

Il modello realizzato per questo problema è solo uno dei tanti possibili; per eseguire eventuali confronti futuri con altri modelli, studiamo la complessità di questo. Sia data un'istanza con $|P|$ punti e con K partizioni da eseguire; allora

- le distanze d_{ij} con $i, j \in P : i < j$ sono numericamente pari alla somma dei primi $|P| - 1$ numeri naturali positivi, ossia

$$\sum_{i=1}^{|P|-1} i = \frac{(|P|-1)|P|}{2} = O(|P|^2), \qquad \text{per } |P| \to +\infty;$$

- le **variabili** sono in totale

$$|P| \cdot K + \frac{(|P|-1)|P|}{2} \cdot K + 2 = \begin{cases} O(|P|^2), & \text{per } |P| \to +\infty \text{ e } K \text{ costante} \\ O(K), & \text{per } K \to +\infty \text{ e } |P| \text{ costante} \end{cases};$$

- i **vincoli** sono in tutto

$$K + |P| + 7 \cdot \frac{(|P|-1)|P|}{2} \cdot K + |P| \cdot K + 2 = \begin{cases} O(|P|^2), & \text{per } |P| \to +\infty \text{ e } K \text{ costante} \\ O(K), & \text{per } K \to +\infty \text{ e } |P| \text{ costante} \end{cases}.$$