

Tripartizione.

E' dato un insieme di punti nel piano Cartesiano, divisi in tre sottoinsiemi. I tre insiemi sono tali che i loro gusci convessi non si sovrappongono.

Si vuole trovare un modo di separare i tre sottoinsiemi con tre semirette aventi l'origine in comune. Tra i vari modi in cui è possibile ottenere la partizione si vuole scegliere quello per cui l'origine risulta:

1. il più possibile vicino all'origine degli assi Cartesiani;
2. il più lontano possibile dall'origine degli assi Cartesiani;
3. all'interno di un quadrato di lato 2 con lati paralleli agli assi e centrato nell'origine degli assi;
4. all'esterno del quadrato suddetto.

Nel terzo e quarto caso si vuole massimizzare la minima distanza tra i punti e le rette che li suddividono.

Formulare il problema, classificarlo e risolvere l'esempio indicato nel seguito.

Esempio.

Punto	x	y
1	-5	12
2	10	8
3	5	5
4	3	0
5	-1	3
6	-2	9
7	2	10
8	10	6

Table 1: Sottoinsieme 1.

Punto	x	y
1	-10	2
2	-4	3
3	-19	15
4	-11	0
5	-7	-3
6	-12	-9

Table 2: Sottoinsieme 2.

Punto	x	y
1	-3	-6
2	-1	-8
3	1	-5
4	-2	-9
5	11	-1
6	2	-9
7	3	-1
8	0	-2
9	1	-8
10	2	-2
11	-3	-6
12	-4	-9

Table 3: Sottoinsieme 3.

Soluzione

I dati del problema sono tre sottoinsiemi $S_j \forall j = 0, \dots, 2$ di punti in posizioni date $(x_i, y_i) \forall i \in S_j \forall j = 0, \dots, 2$.

Il problema richiede di determinare tre rette nel piano ed il loro punto comune di intersezione. Servono quindi tre variabili continue e libere a_j , b_j e c_j che rappresentano i coefficienti dell'equazione $a_j x + b_j y + c_j = 0$ per ogni retta $j = 0, \dots, 2$. Utilizzare indici da 0 a 2 anziché da 1 a 3 è utile per poter imporre condizioni cicliche, utilizzando gli indici modulo 3.

Affinché tali variabili assumano valori che rappresentano effettivamente l'equazione di una retta, occorre che non siano tutti e tre nulli. Ciò si può imporre con la condizione di normalizzazione $a_j^2 + b_j^2 = 1$ per ogni retta $j = 0, \dots, 2$.

Per rappresentare il punto P di intersezione usiamo altre due variabili continue e libere, x_P e y_P , e imponiamo che $a_j x_P + b_j y_P + c_j = 0$ per ogni retta $j = 0, \dots, 2$.

Per imporre che le tre rette partizionino come richiesto i tre sottoinsiemi di punti, bisogna imporre che le coordinate dei punti dati, sostituiti nell'equazione delle rette, producano valori sempre non positivi in un caso e sempre non negativi nell'altro, il che equivale ad imporre che tutti i punti di un sottoinsieme cadano da un lato della retta e tutti i punti dell'altro sottoinsieme cadano dall'altro lato.

Questo vincolo si può imporre grazie all'assunzione che i gusci convessi dei tre insiemi non si sovrappongano. Di conseguenza le ampiezze dei tre angoli tra le semirette sono sempre inferiori a 180 gradi e tutti i punti di ogni sottoinsieme devono stare dalla stessa parte rispetto a due delle rette.

Per ogni retta sono solo due i sottoinsiemi interessati; i punti del terzo sottoinsieme possono quindi cadere da entrambe i lati indifferentemente. Ogni retta ammette due rappresentazioni equivalenti, che differiscono per il segno dei coefficienti e corrispondono al verso entrante in P o uscente da P . Ai fini dell'esercizio è indifferente scegliere una o l'altra delle rappresentazioni, purché sia la stessa per ogni retta.

Imponiamo ad esempio che per ogni retta j valgano le condizioni:

$$a_j x_i + b_j y_i + c_j \leq 0 \quad \forall i \in S_j$$

$$a_j x_i + b_j y_i + c_j \geq 0 \quad \forall i \in S_{j+1}$$

dove gli indici di ogni sottoinsieme S sono da intendere modulo 3.

Per quanto riguarda la funzione obiettivo, ne vengono proposte due. Nel primo caso, si vuole minimizzare la distanza dall'origine, cioè

$$\text{minimize } z = \sqrt{x_P^2 + y_P^2}.$$

La radice quadrata si può omettere, dal momento che minimizzare il quadrato della distanza ha lo stesso effetto di minimizzare la distanza.

Nel secondo caso si vuole massimizzare la stessa distanza, cioè

$$\text{maximize } z = \sqrt{x_P^2 + y_P^2}$$

e anche in questo caso si può omettere la radice quadrata.

Nelle versioni 3 e 4 dell'esercizio la funzione obiettivo da massimizzare è la minima distanza tra punti e rette. Si introduce pertanto una variabile ausiliaria δ per linearizzare la funzione-obiettivo max-min e si massimizza δ , modificando i vincoli precedenti:

$$a_j x_i + b_j y_i + c_j \leq -\delta \quad \forall i \in S_j$$

$$a_j x_i + b_j y_i + c_j \geq \delta \quad \forall i \in S_{j+1}.$$

Per restringere il punto P all'interno del quadrato di lato 2 e centro in O , basta imporre limiti superiori (pari a 1) e inferiori (pari a -1) alle due coordinate.

Per restringere il punto P all'esterno del quadrato è necessario utilizzare vincoli disgiuntivi. I vincoli devono poter essere disattivati, dato che non possono essere tutti soddisfatti contemporaneamente. Per disattivarli è possibile introdurre una variabile binaria w per ogni vincolo ed una costante M grande abbastanza, in modo da ottenere il sistema seguente:

$$\begin{cases} x_P \leq -1 + Mw_1 \\ x_P \geq 1 - Mw_2 \\ y_P \leq -1 + Mw_3 \\ y_P \geq 1 - Mw_4 \end{cases}$$

Ognuno dei vincoli impone che il punto P sia fuori dal quadrato in una delle quattro direzioni possibili, ed è imposto quando la corrispondente variabile binaria w vale 0. Quando invece $w = 1$, il vincolo risulta sempre soddisfatto indipendentemente dalla posizione di P , purché M sia abbastanza grande.

La condizione che il punto P cada fuori dal (o sul contorno del) quadrato, cioè che almeno uno dei quattro vincoli disgiuntivi sia soddisfatto, è quindi

$$w_1 + w_2 + w_3 + w_4 \leq 3.$$

Il problema è chiaramente di programmazione non-lineare. Nella versione 1 la funzione obiettivo da minimizzare è convessa ed il minimo è unico. Nella versione 2 possono esistere minimi locali, poiché si massimizza una funzione obiettivo convessa. Nella versione 3 e 4 la funzione obiettivo è convessa; nella versione 3 i vincoli sono convessi, nella versione 4 no.

Variante migliorata. Il problema si può risolvere senza imporre l'ipotesi che i gusci convessi degli insiemi di punti dati non si sovrappongano.

A questo scopo si può definire una direzione centrale per ogni cluster, cioè la direzione di una semiretta che esce dal punto origine (x_P, y_P) e attraversa il cluster. Si considerino quindi i versori relativi a tutti i punti dati, cioè i vettori di lunghezza unitaria uscenti dal punto origine e orientati da esso verso ciascun punto dato. Esiste una partizione corretta dei punti dati se il prodotto scalare tra i versori dei punti di un cluster k e la direzione centrale del cluster k è maggiore o uguale al prodotto scalare tra i versori dei punti degli altri cluster e la direzione centrale del cluster k .

Nei modelli relativi a questa versione migliorata, nelle varianti 3 e 4, si è massimizzata per comodità la differenza tra tali prodotti scalari, anziché la distanza punto-retta.