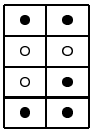
L'algoritmo di Hoshen-Kopelman

Lalgoritmo di *Hoshen-Kopelman* (che d'ora in poi verrá chiamato algoritmo HK76, [HK76]) é un esempio di **cluster multiple labeling technique**. Il reticolo viene visitato sito per sito per colonne partendo dallo spigolo in alto a sinistra per arrivare a quello in basso a destra. Per spiegare meglio il suo funzionamento a linee generali é utile analizzare il piccolo reticolo in figura, i pallini neri indicano siti occupati.



Quando si incontra un sito occupato che non é connesso ad altri siti occupati sopra o a sinistra inizia un nuovo *cluster* e viene assegnato al sito un nuovo *label*[5], per l'assegnazione dei *label* si partirá dal valore uno. Invece, quando c'é un primo vicino sopra o a sinistra occupato (uno solo dei due) il sito in esame prende il valore del sito primo vicino occupato, analogamente se i suoi primi vicini sono entrambi occupati ma con lo stesso *label*. Dopo sette passi il nostro reticolo appare cosí:

1	1
	3
2	?

[5] numero che identifica i cluster, in linea di principio siti appartenenti a cluster diversi hanno *label* diversi

L'ottavo sito ha due primi vicini occupati con valori (*label*) differenti, quale viene scelto dei due? Si sceglie il minore ma bisogna fare attenzione perché in realtá il sito con valore 3 (anzi, tutto il "cluster 3") appartiene allo stesso cluster. Cambiare il valore di tutti i 3 comporterebbe un duro lavoro, soprattutto per grossi cluster. L'idea dellalgoritmo HK76 é di non assegnare nuovi valori ai siti (nell'esempio a quello identificato dal 3), ma prendere nota che 2 e 3 sono lo stesso cluster. In pratica questo viene fatto usando un vettore chiamato *Label of Label* (*LofL*) che contiene tutta l'informazione necessaria sui label dei cluster: per un *good label* (come 2 nel caso precedente) memorizza la taglia (momentanea) del cluster, per *bad label* (come 3 nel caso precedente) memorizza qual é il vero cluster label a cui questo label appartiene. Questa distinzione viene fatta attraverso il segno del numero intero contenuto nella componente in esame di LofL. Funziona in questo modo:

- $LofL(goodlabel) =$ taglia
- $LofL(badlabel) =$ indrizzo del label di riferimento[6]

Questo compito viene svolto da **HKclass** che preso in ingresso un label (e naturalmente *LofL*) restituisce il *good label* corrispondente. Di fatto HKclass si autorichiama finché il label non é positivo.

Alla fine l'algoritmo HK (a meno che non venga fatta una rilabelizzazione successiva) non garantisce che tutti i siti di un fissato cluster abbiano lo stesso valore (mentre siti di cluster diversi non possono avere lo stesso valore) ma restituisce in modo corretto le taglie dei cluster (unica quantitá a cui siamo interessati per la nostra analisi).

Osservazione: la labelizzazione dei siti e la classificazione dei cluster possono essere svolti simultaneamente alla creazione del reticolo, vantaggi: si fa la simulazione di un grande reticolo senza doverlo memorizzare tutto. Dato che il reticolo viene visitato colonna per colonna e servono informazione solo riguardo ai primi vicini sopra e a sinistra rispetto al sito in esame, si possono tenere in memoria solo due righe alla volta (o piú nel caso di condizioni periodiche).

Questo vantaggio é particolarmente utile in dimensioni maggiori di due, se si lavora in d dimendioni si memorizzano oggetti in $d-1$ dimensioni .

[6] non necessariamente é il *good label*, possono essere necessari piú passi per arrivare al valore corretto.