Corso di Laurea Specialistica in Ingegneria Gestionale
a.a. 2005-06

Sistemi Informativi per le Decisioni LS

# Clustering Association Rules

## Brian Lent - Arun Swami - Jennifer Widom

Prof. Marco Patella

**GRUPPO IX**

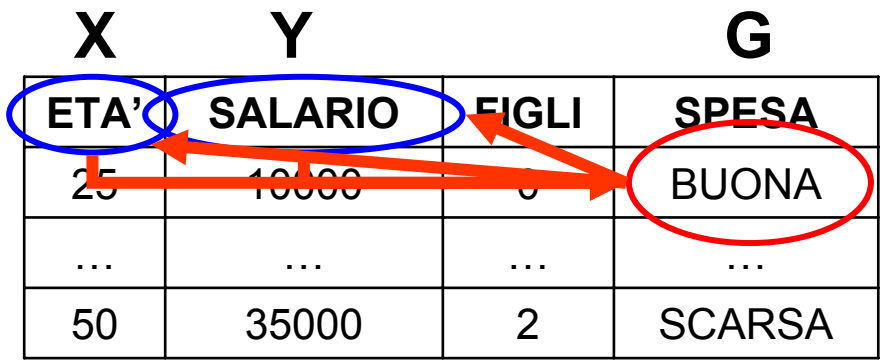
Angelini Andrea

Menenti Alice

Della Vittoria Lucia

# Il Framework ARCS

- Ambiti applicativi:
  - Segmentazione
  - Classificazione
- Input:
  - Large Data Base
  - Valore dell'attributo di segmentazione (es.: G = 'BUONA')
  - 2 attributi significativi per analizzare le caratteristiche del DB (X, Y)
  - Numero di bin (parametro)
- Output:
  - Cluster di regole associative costruiti su X e Y



# Cos'è un cluster di regole associative

Dalle seguenti regole associative:

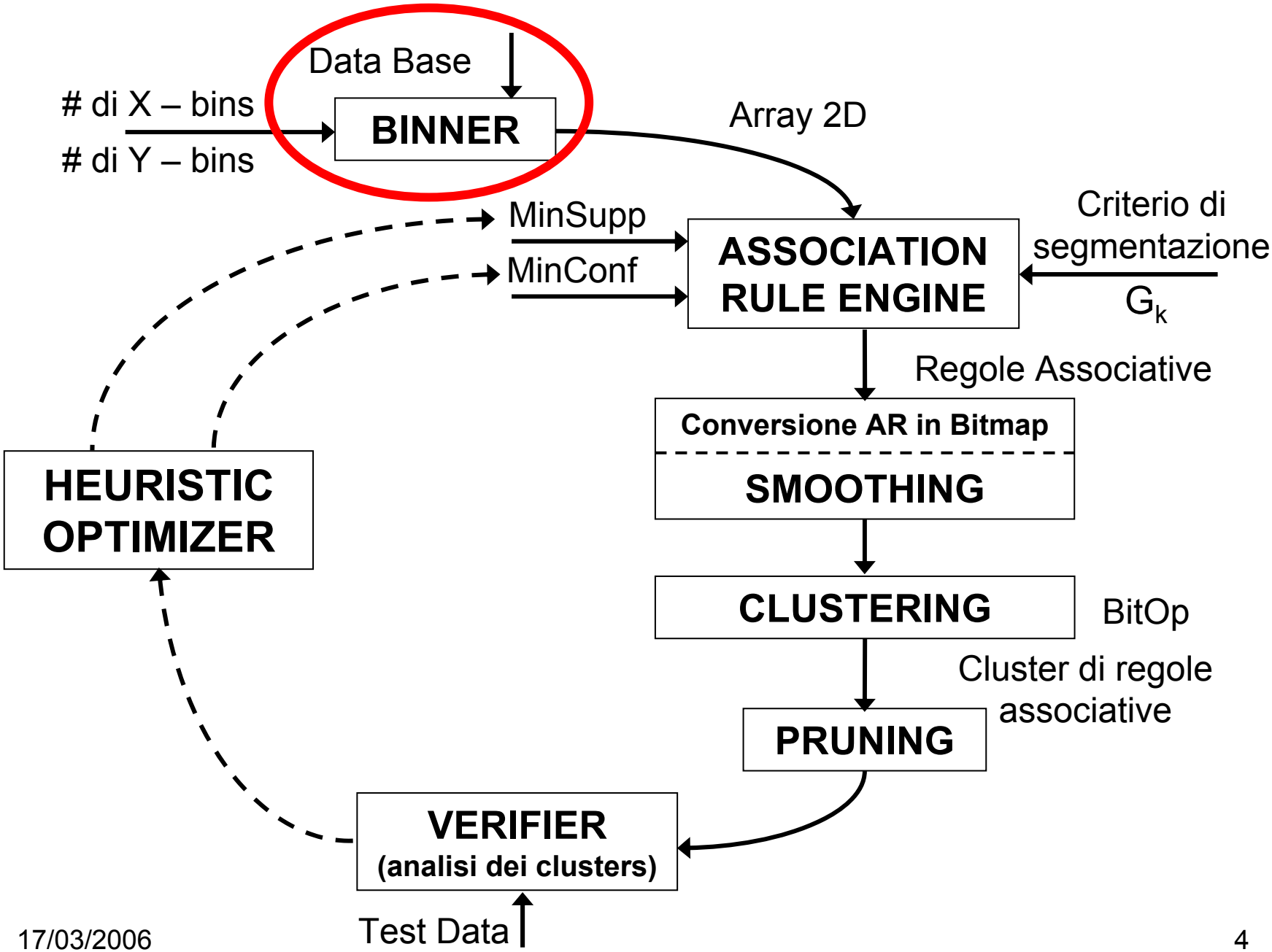
(Età = 40) Λ (Salario = \$42,350) $\Rightarrow$ (Spesa = BUONA)
(Età = 41) Λ (Salario = \$57,000) $\Rightarrow$ (Spesa = BUONA)
(Età = 41) Λ (Salario = \$48,750) $\Rightarrow$ (Spesa = BUONA)
(Età = 40) Λ (Salario = \$52,600) $\Rightarrow$ (Spesa = BUONA)

Si determina il cluster di regole associative:

(40 ≤ Età < 42) Λ (\$40,000 ≤ Salario < \$60,000) $\Rightarrow$ (Spesa = BUONA)

Generalizzando:

$$\mathbf{(X_1 \leq X < X_2) \Lambda (Y_1 \leq Y < Y_2) \Rightarrow (G = G_K)}$$


Data Base
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

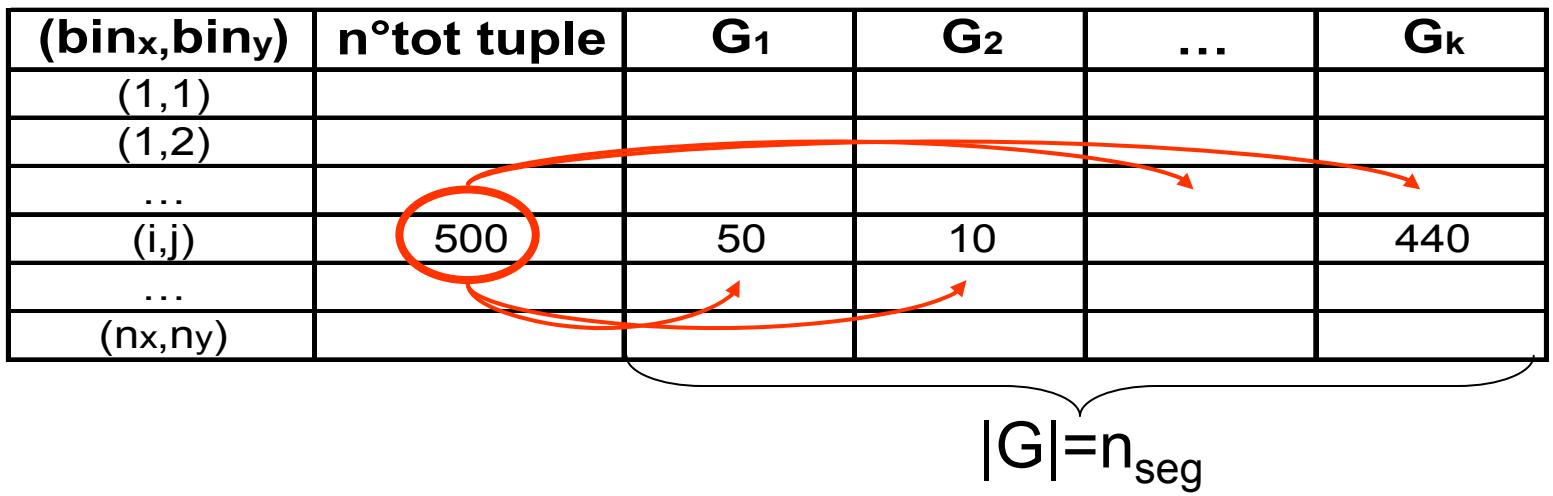
# Binner

- Gli attributi X e Y sono quantitativi

  ⟶ è necessario scegliere il numero di bin con cui effettuare il partizionamento del loro dominio
- Il numero di bin è un **parametro** scelto dall'utente!!
- Partizionamento domini con criterio equi-width
- Successiva costruzione dell'**array 2D**, rappresentazione sintetica delle caratteristiche del DB

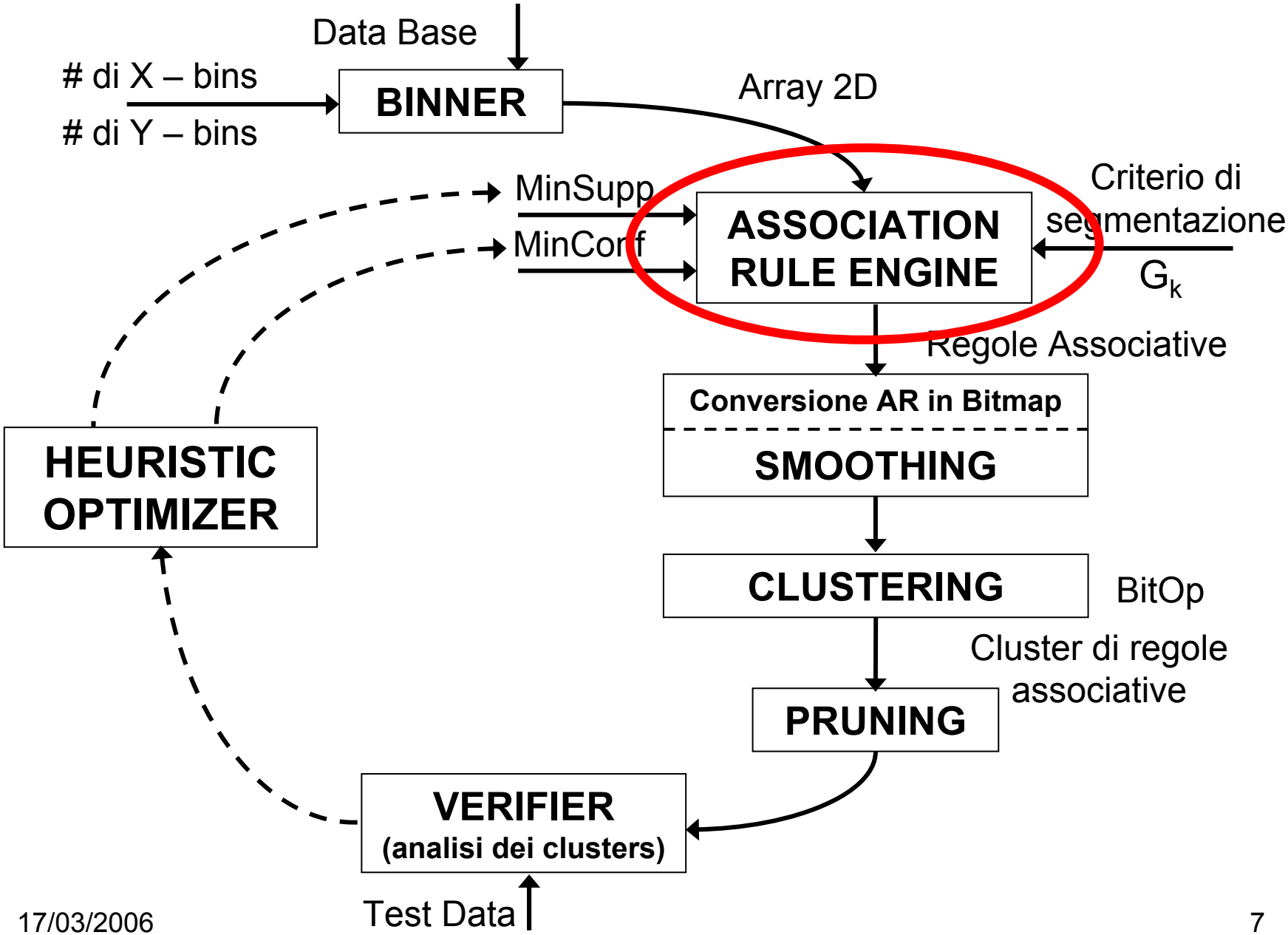
# Array 2D

| ($bin_x$,$bin_y$) | n°tot tuple | $G_1$ | $G_2$ | … | $G_k$ |
|---|---|---|---|---|---|
| (1,1) | | | | | |
| (1,2) | | | | | |
| … | | | | | |
| (i,j) | 500 | 50 | 10 | | 440 |
| … | | | | | |
| (nx,ny) | | | | | |




- Ogni bin viene associato ad un numero intero progressivo
- La dimensione dell'array 2D è $n_x * n_y * (n_{seg}+1)$
- L'array 2D può essere memorizzato in memoria centrale

  → cambiando il valore di $G_k$ per effettuare una nuova segmentazione, non è necessario rileggere i dati da DB


Data Base
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

# Association rule engine (i)

Array 2D

| ($bin_x$,$bin_y$) | n°tot tuple | $G_1$ | $G_2$ | … | $G_k$ |
|---|---|---|---|---|---|
| (1,1) | | | | | |
| (1,2) | | | | | |
| … | | | | | |
| (i,j) | 500 | 50 | 10 | | 440 |
| … | | | | | |
| ($n_x$,$n_y$) | | | | | |

MinSupp 25%

MinConf 80% ⇨ Supporto $|(i, j, G_k)| / N = 440/1500 \approx 30\%$

N 1500

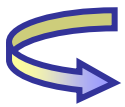
Confidenza $|(i, j, G_k)| / |(i, j)| = 440/500 \approx 88\%$

GenAssociationRules() restituisce la coppia (i,j) che identifica la regola associativa $(X = i) \wedge (Y = j) \Rightarrow G_k$

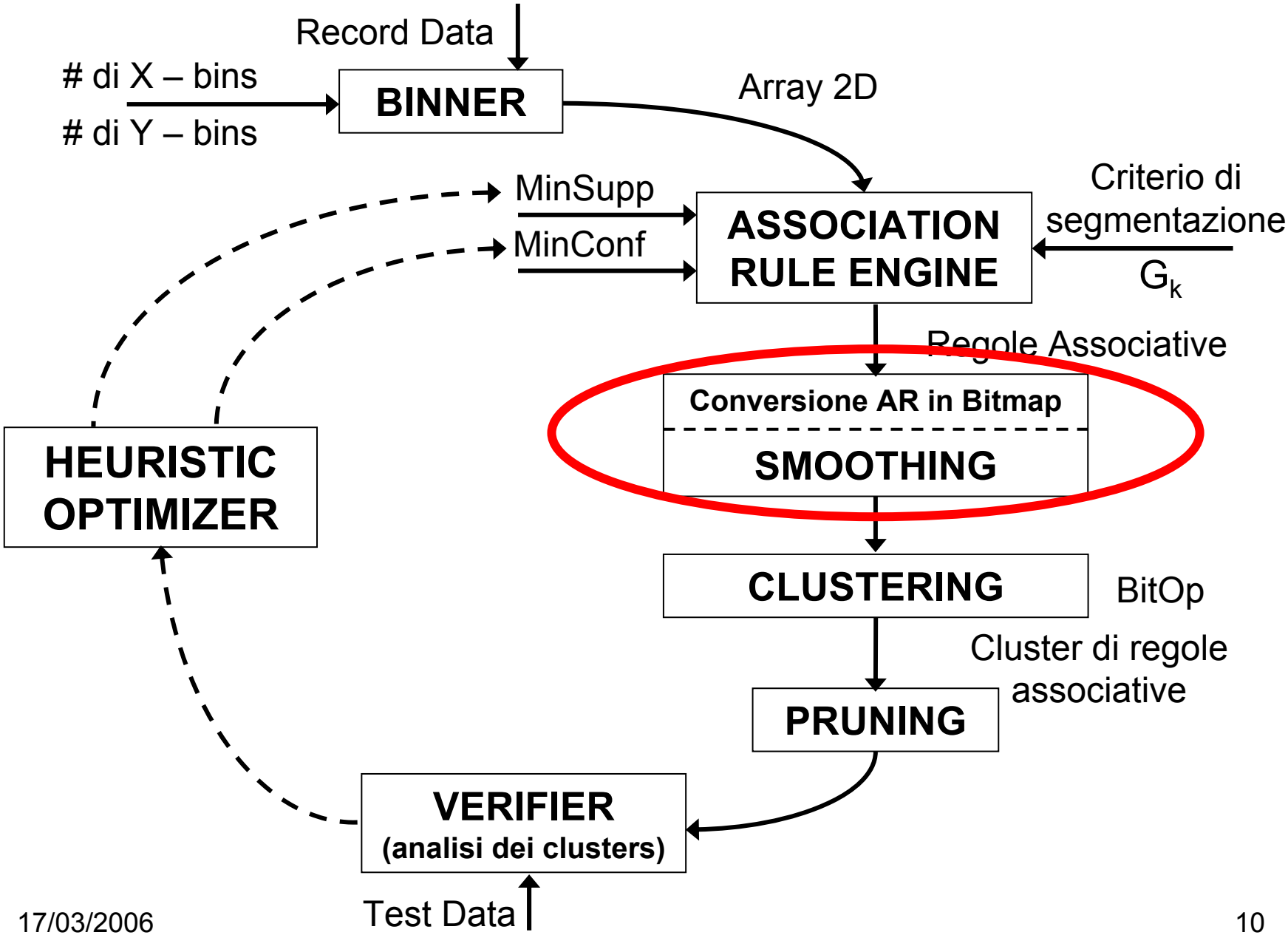
# Association rule engine (ii)

Quali **vantaggi** offre l'Association Rule Engine applicato all'array 2D?

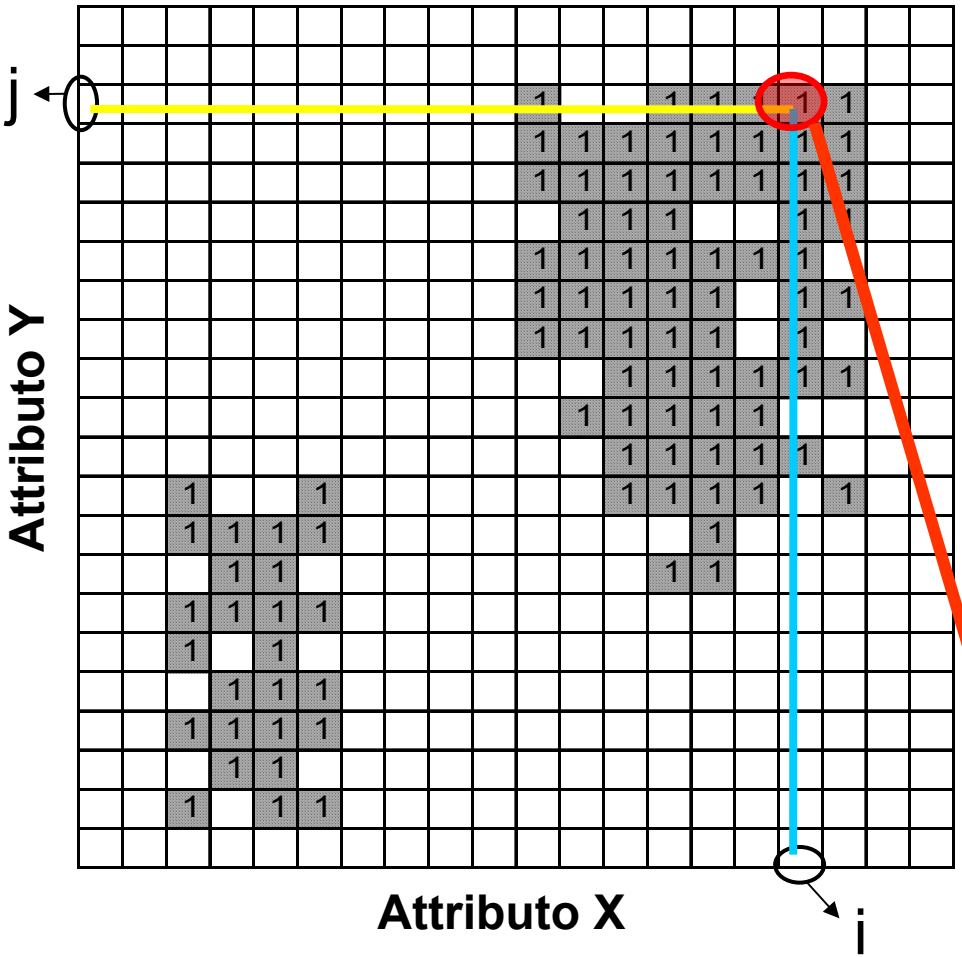
- Tutte le regole associative per $G_k$ in un'unica scansione
- Array 2D memorizzabile in memoria centrale



parametri MinSupp e MinConf modificabili senza riesaminare le tuple del DB


Record Data
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

# Conversione AR in Bitmap Grid




Che cos'è una **Bitmap Grid?**

- Ad ogni coppia (i,j) restituita dalla procedura GenAssociationRules() viene assegnato il valore "1"
- Le altre caselle contengono valore "0"

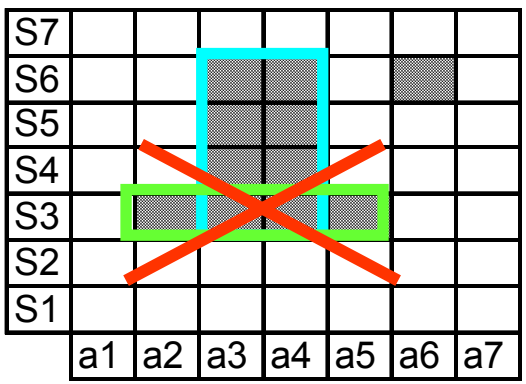
$$(X = i) \wedge (Y = j) \Rightarrow G_k$$

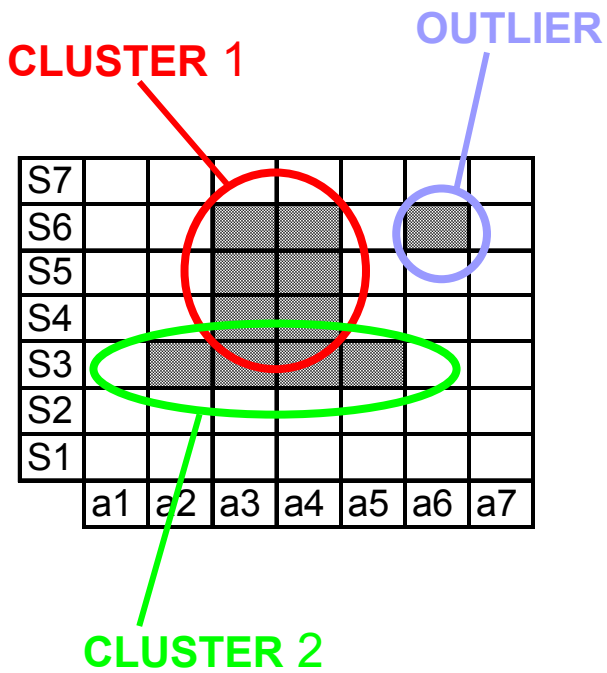
# Come utilizzare la Bitmap Grid?

Dalla Bitmap Grid si vuole individuare il minimo numero di cluster:

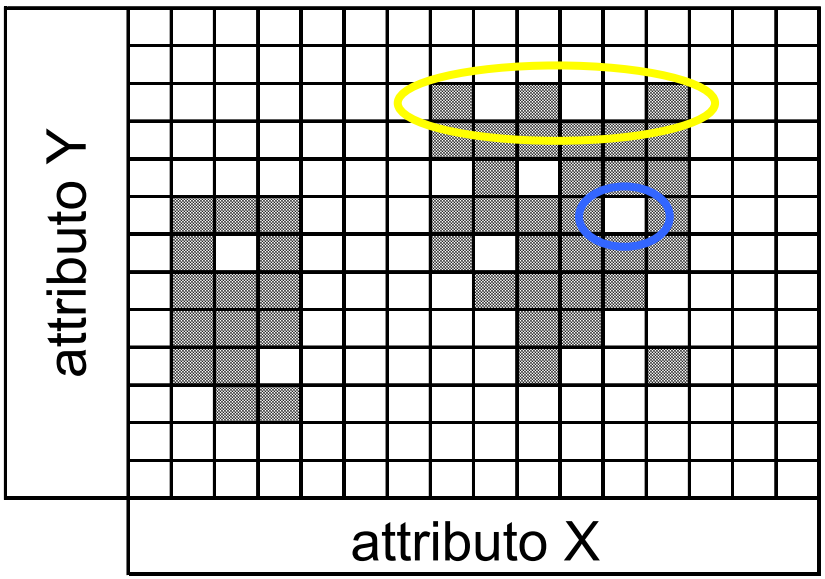
- di forma rettangolare
- non sovrapposti

$$(X = i) \wedge (Y = j) \Rightarrow G_k$$



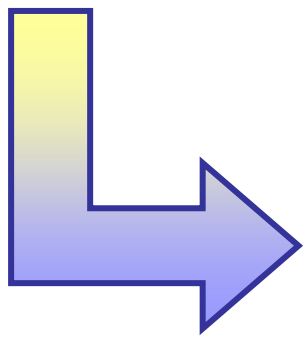



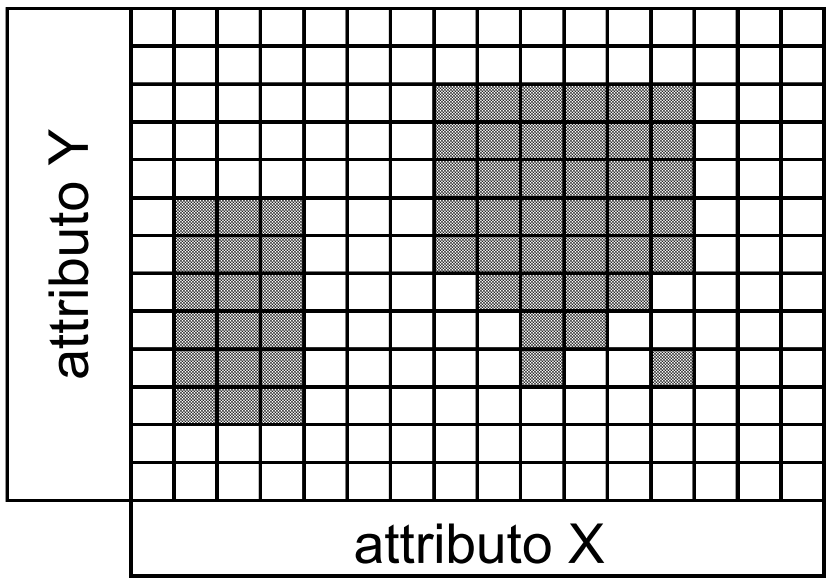
# Smoothing

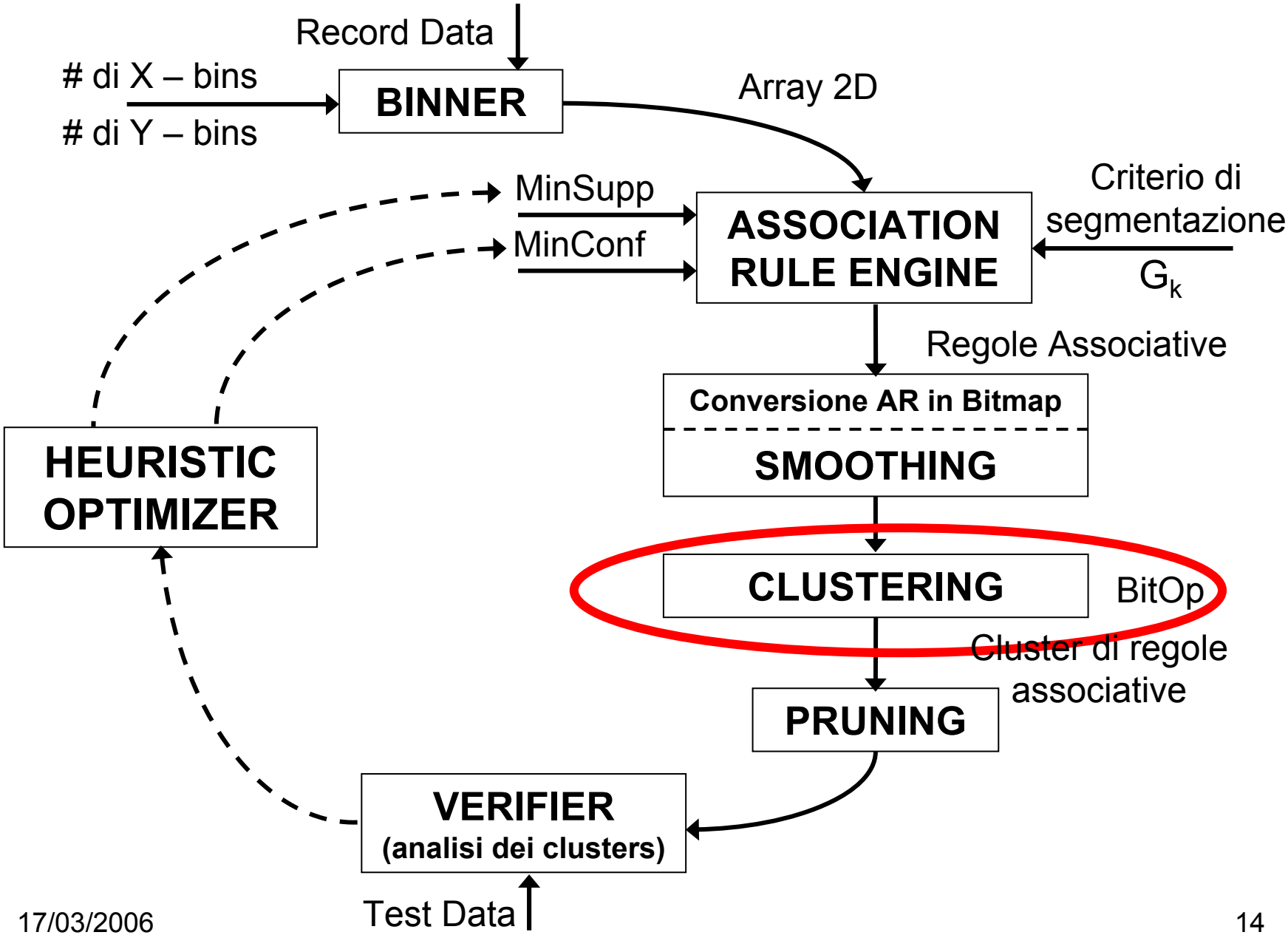



Tipicamente, le celle significative nella Bitmap Grid tendono a formare regioni con:

- confini frastagliati
- vuoti all'interno (rumore)



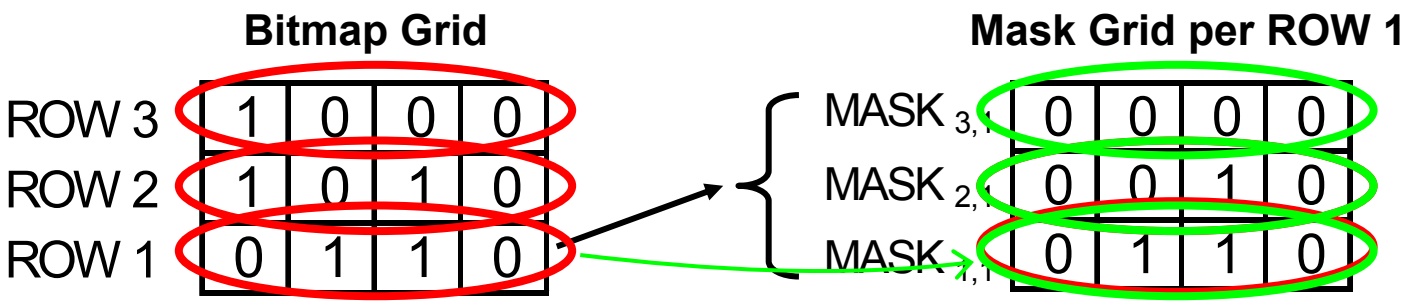



Record Data
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

# Clustering (i)

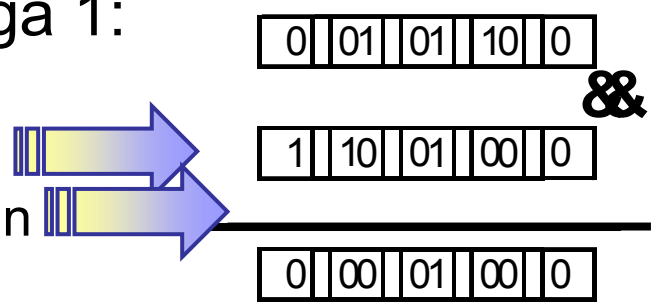
## Algoritmo BitOp

**Input:**

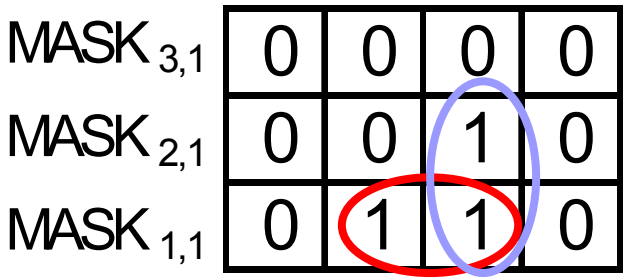



**Procedura:**

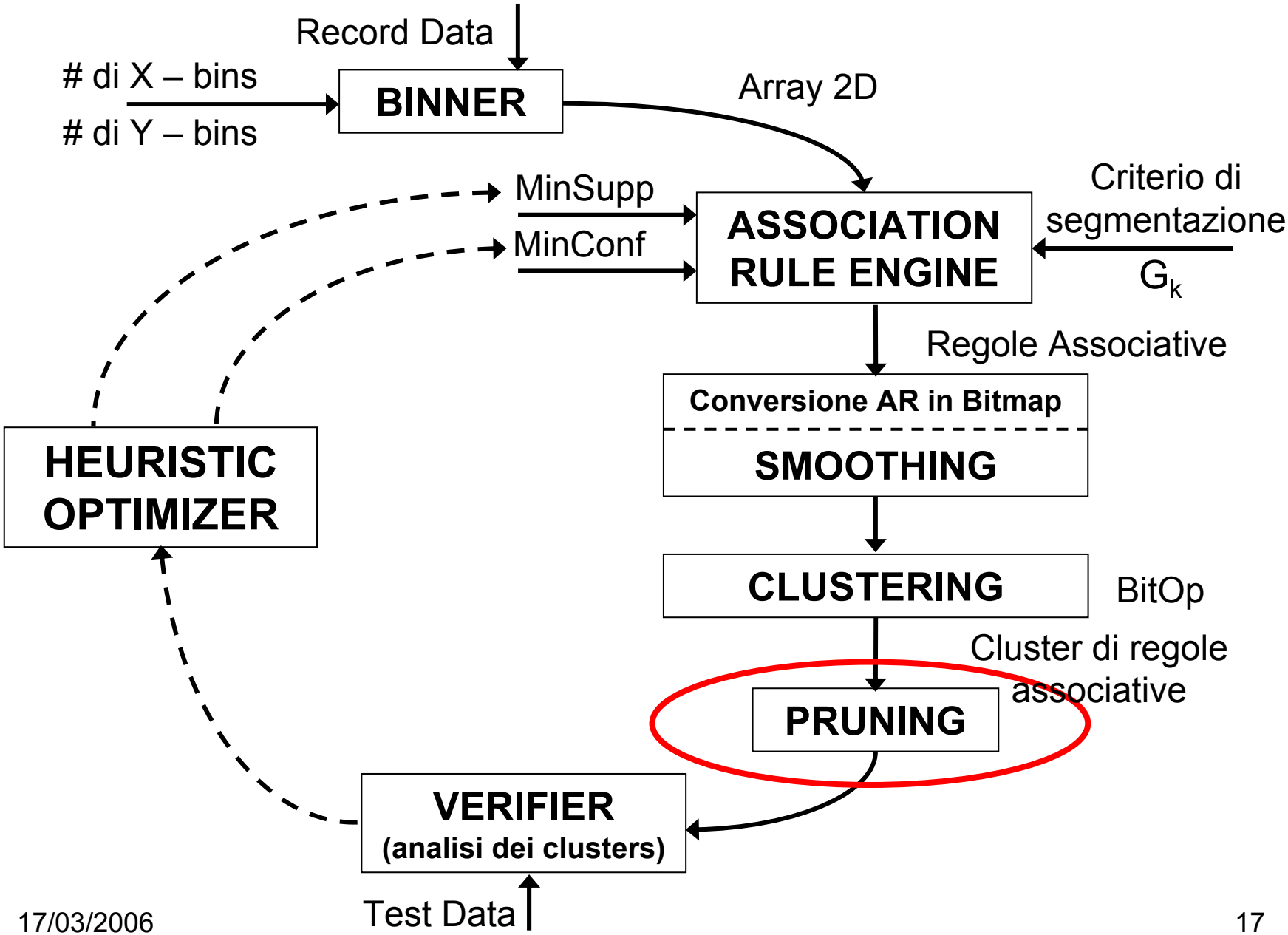
- Per ogni riga della Bitmap Grid si crea una Mask Grid
- Costruzione Mask Grid per riga 1:
  1) MASK $_{1,1}$ = ROW 1
  2) MASK $_{2,1}$ = MASK $_{1,1}$ & ROW 2
  3) MASK $_{n,1}$ = MASK $_{(n-1),1}$ & ROW n



# Clustering (ii)



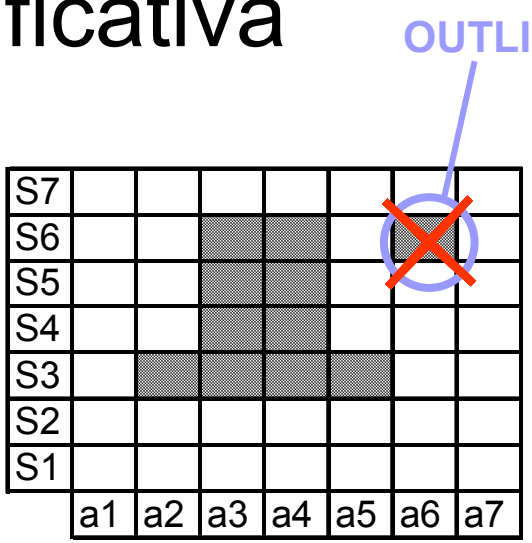

- Minimo numero di cluster di massima dimensione
- Complessità $\mathbf{O(\Sigma_{S \epsilon C} |S|)}$,
  $\boldsymbol{C}$ = set finale di clusters individuati
- Approccio euristico
- Approssimazione efficiente di un clustering ottimale

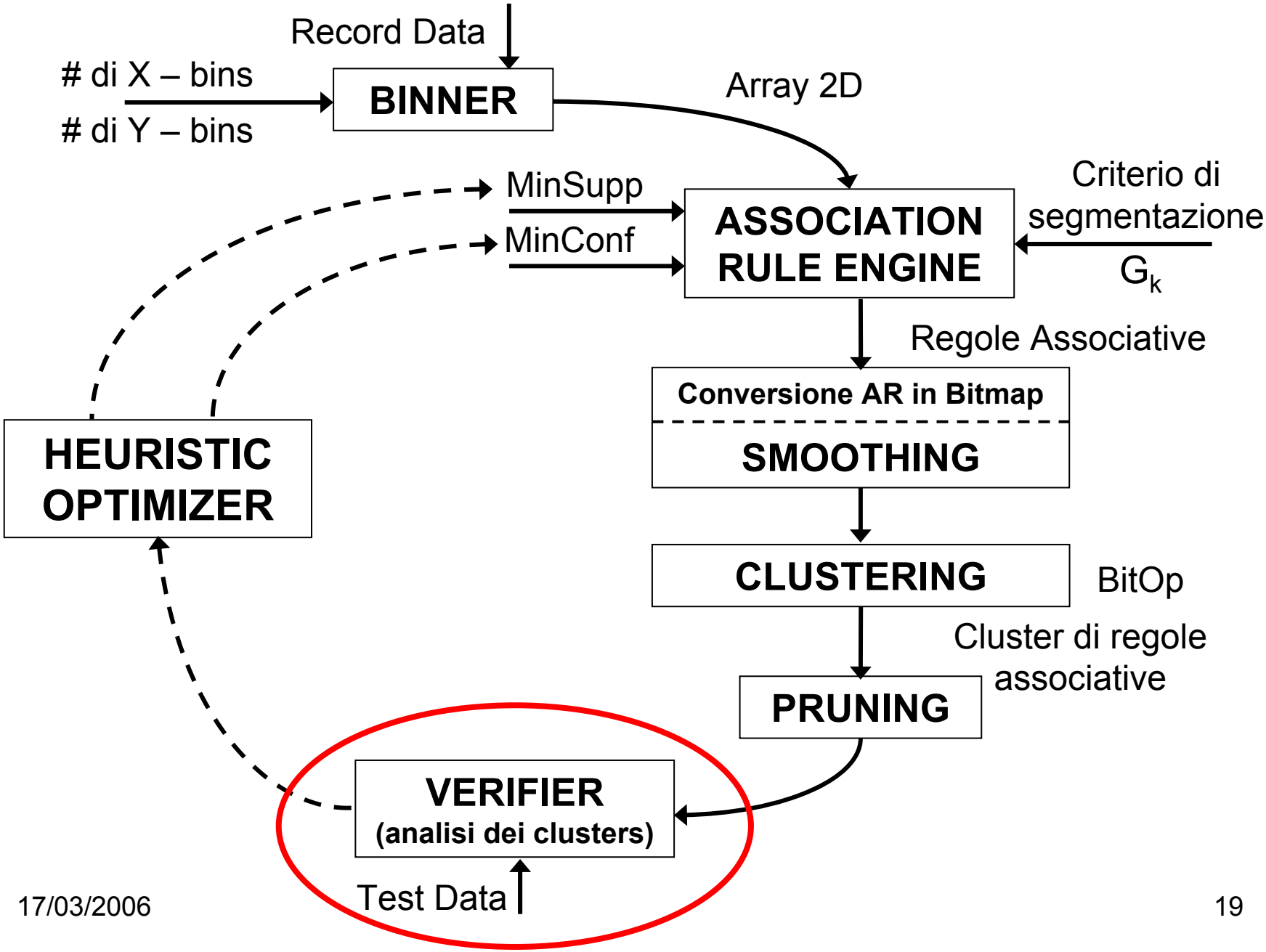

Record Data
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

# Pruning

Tipicamente i clusters di dimensioni <1%della superficie della Bitmap Grid risultano non rilevanti per una segmentazione significativa

Eliminazione degli **outliers**




Record Data
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
HEURISTIC OPTIMIZER
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data

# Verifier - qualità di segmentazione

Utilizzo del principio **MDL** (Minimum Description Lenght):

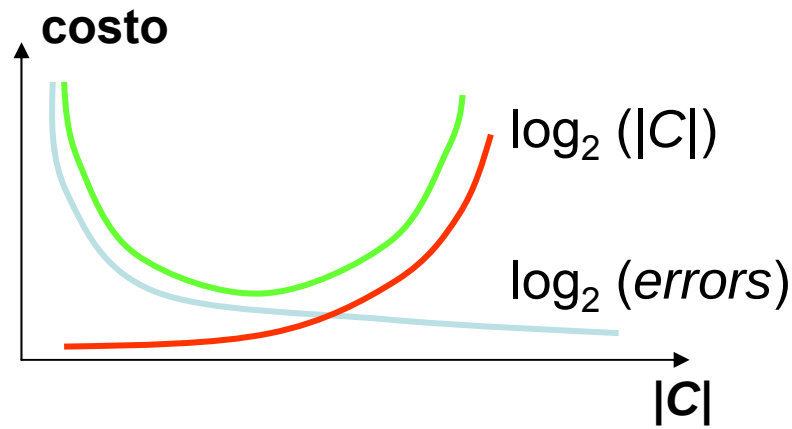
Obiettivo: min costo totale di descrizione dei clusters

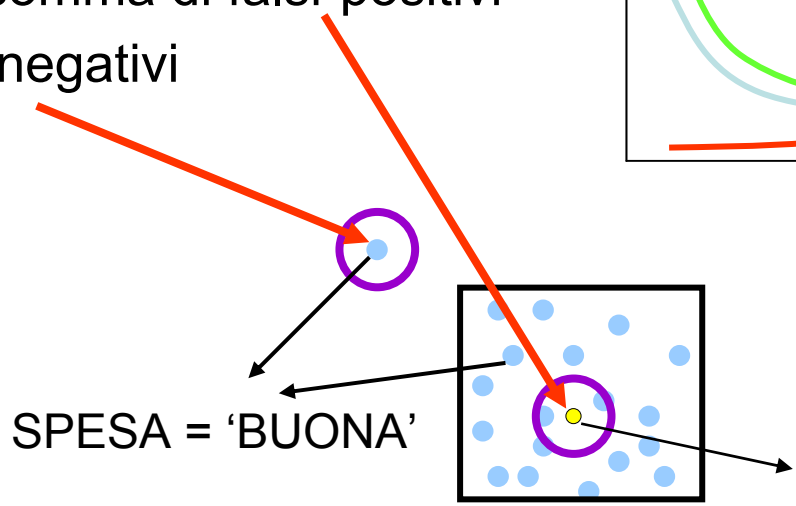
$$\textbf{costo} = \mathbf{w_c}\ \log_2(|C|) + \mathbf{w_e}\ \log_2(\textit{errors})$$

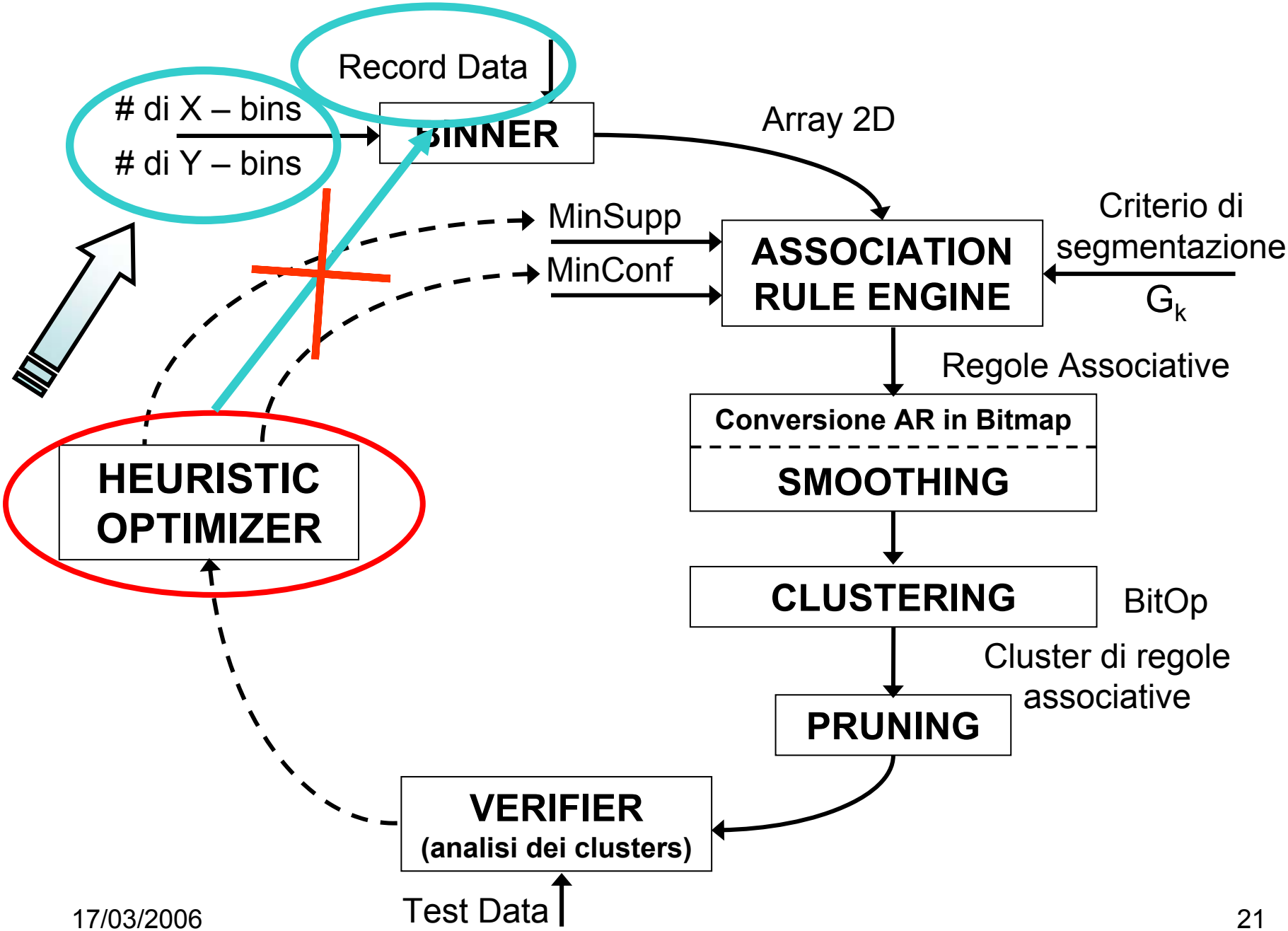
$|C|$ = numero di clusters

$\mathbf{w_c}$, $\mathbf{w_e}$ = pesi (scelti dall'utente)

***errors*** = somma di falsi-positivi e falsi-negativi



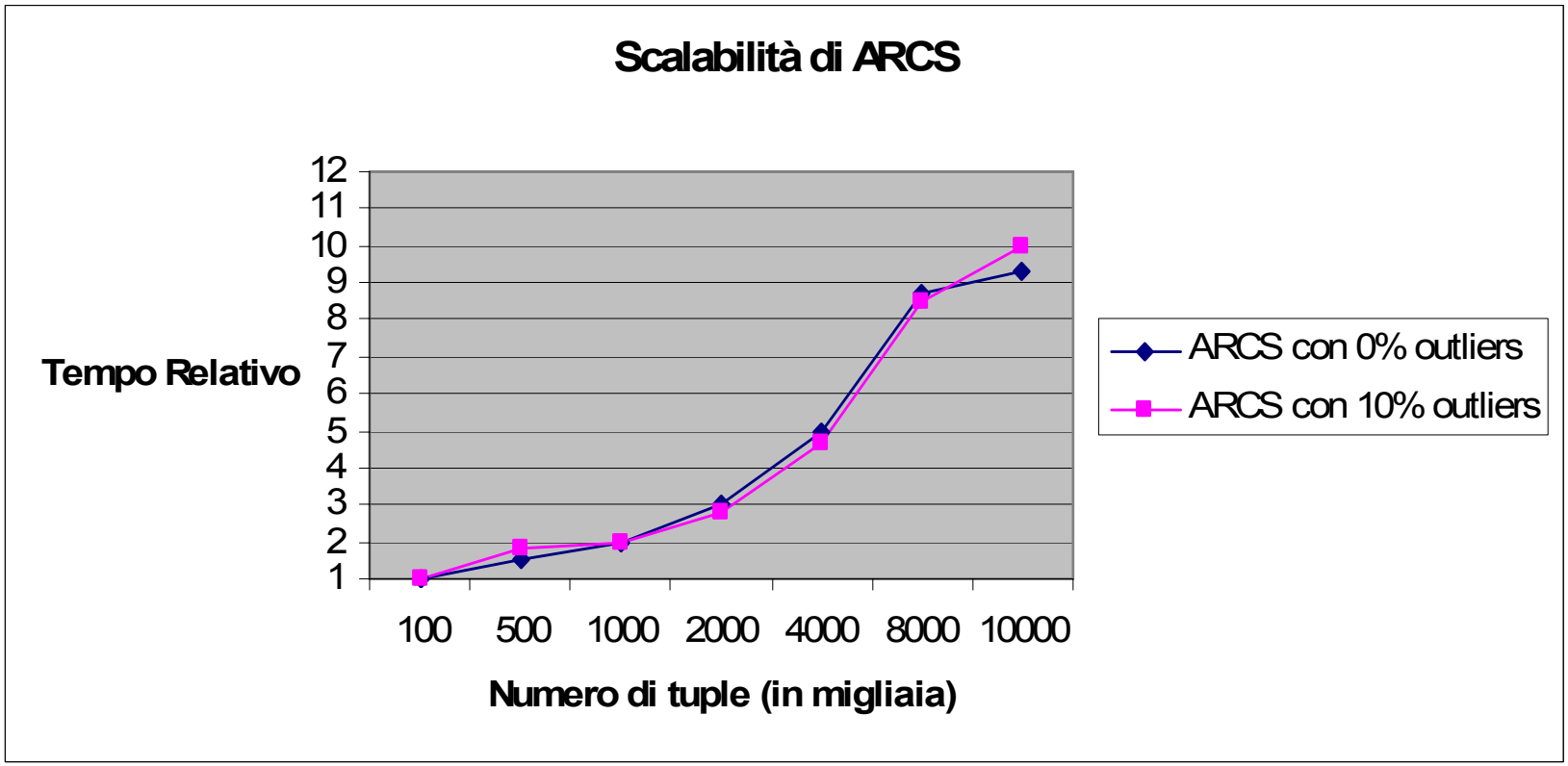

Record Data
# di X – bins
# di Y – bins
BINNER
Array 2D
MinSupp
MinConf
ASSOCIATION RULE ENGINE
Criterio di segmentazione
$G_k$
Regole Associative
Conversione AR in Bitmap
SMOOTHING
CLUSTERING
BitOp
Cluster di regole associative
PRUNING
VERIFIER
(analisi dei clusters)
Test Data
HEURISTIC OPTIMIZER

# Confronto ARCS e C4.5(i)

- Comparazione dei risultati ottenuti su entrambi i datasets con ARCS e C4.5 su un Data Set sintetico con aggiunta di 5% di rumore e outlier
- C4.5 permette di costruire accurati decision trees per la classificazione dei dati

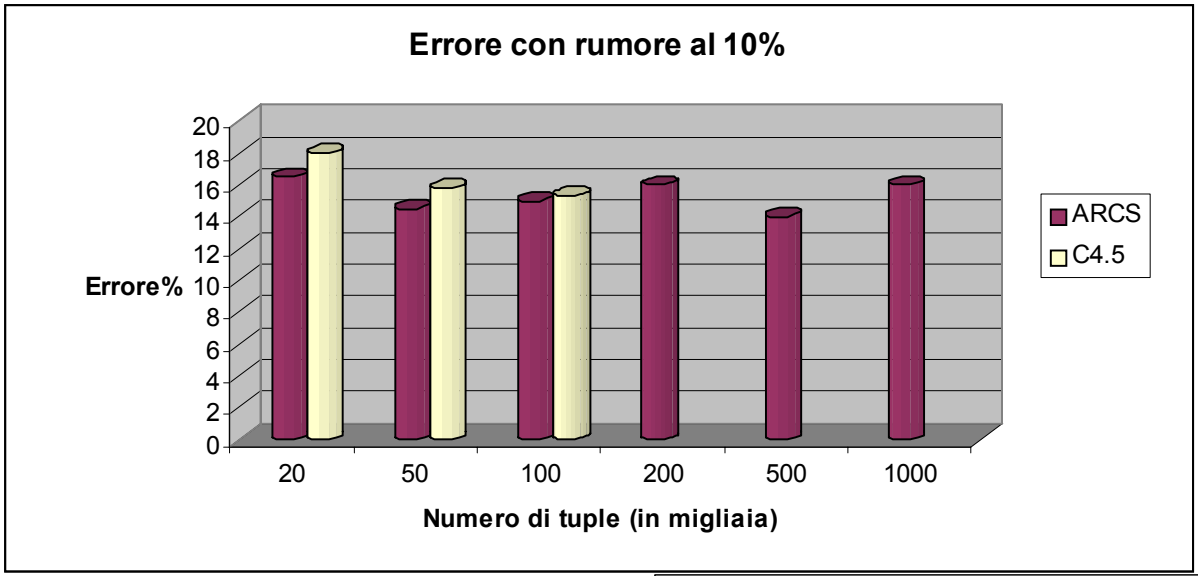
| | ARCS | | C4.5 | |
|---|---|---|---|---|
| D | σ = 0% | σ = 10% | σ = 0% | σ = 10% |
| 20000 | 27 | 28 | 14 | 27 |
| 50000 | 37 | 43 | 81 | 190 |
| 100000 | 42 | 41 | 210 | 893 |
| 200000 | 45 | 47 | 650 | 3K |
| 500000 | 56 | 62 | 4K | 20K |
| 1 milione | 80 | 82 | 15K | 86K |
| 2 milioni | 123 | 117 | n/a | n/a |
| 4 milioni | 203 | 192 | n/a | n/a |
| 8 milioni | 367 | 349 | n/a | n/a |
| 10 milioni | 420 | 426 | n/a | n/a |

# Confronto ARCS e C4.5 (iii)




Il tempo di esecuzione di ARCS aumenta al massimo linearmente con la dimensione del DB

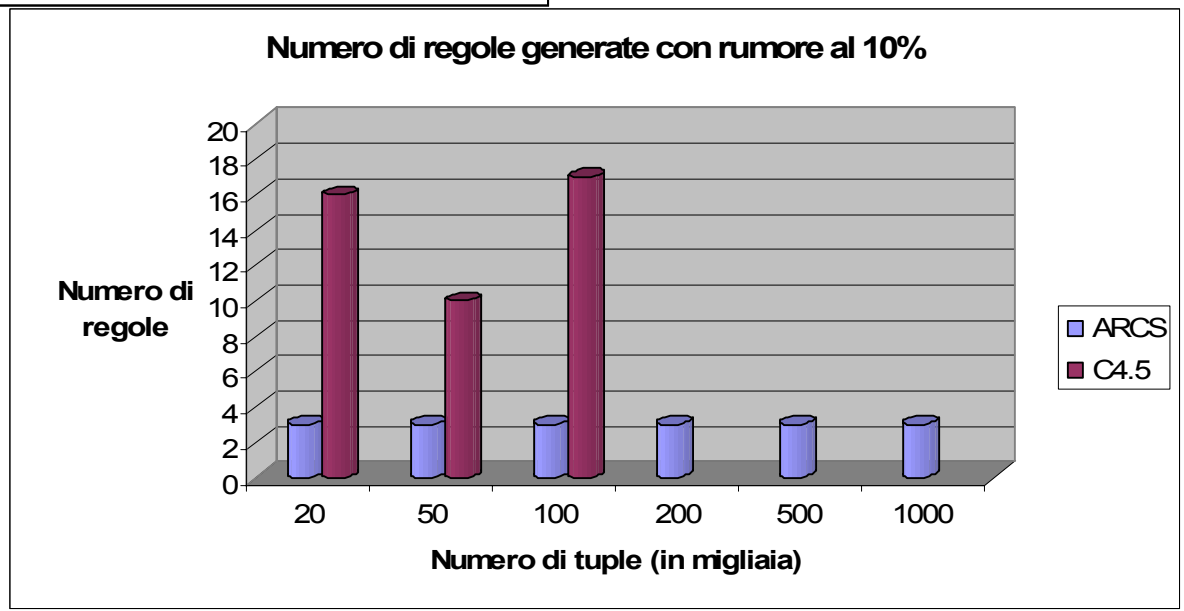
# Confronto ARCS e C4.5 (ii)




ARCS garantisce migliori prestazioni in presenza di rumore grazie a:

- smoothing
- pruning
- ricerca di MinSupp e MinConf ottimali

Riuscire ad ottenere poche regole associative clusterizzate è fondamentale per l'utente che dovrà poi utilizzarle



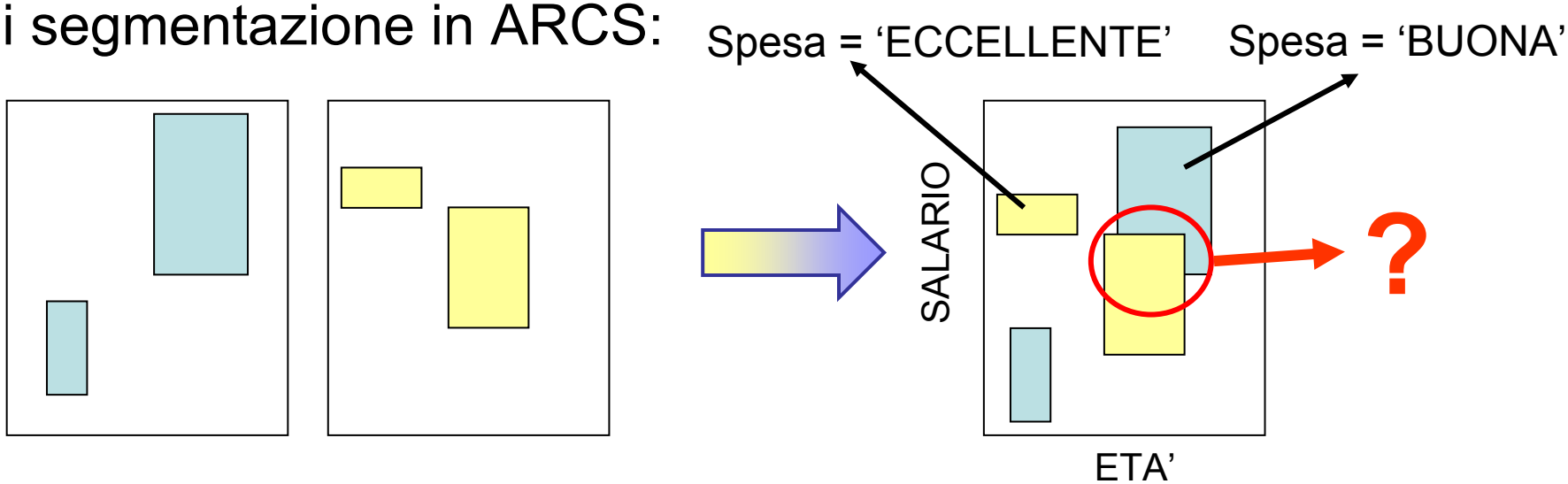
# Criticita’

1) Dimensione dei bin
   – Ogni volta in cui il numero di bin viene cambiato è necessario rileggere tutti i dati
2) Cluster di regole associative solo
   – a 2 dimensioni
   – con attributi quantitativi $(X = i) \wedge (Y = j) \Rightarrow G_k$
3) Overlapping dei segmenti
   – Sovrapponendo griglie bitmap costruite per valori diversi dell’attributo di label, potrebbero sovrapporsi celle di “1”

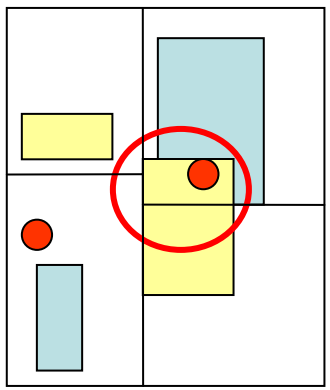
   ⟶ classificazione non univoca

# Criticita’(3)

Segmenti ottenuti per due diversi valori dell’attributo di segmentazione in ARCS:




DT divide in sequenza, ortogonalmente i due attributi



# Conclusioni

- Il data set su cui è stato valutato ARCS ne favorisce l'efficienza
- Un'attenta analisi del paper ha evidenziato le criticità precedentemente riportate
- Una nostra proposta di confronto fra ARCS e DT

# Conclusioni

Caratteristiche del DT proposto:

–Partiziona il dominio degli attributi in bin

–Memorizza la tabella di contingenza 3D sugli attributi partizionati in bin