Clustering

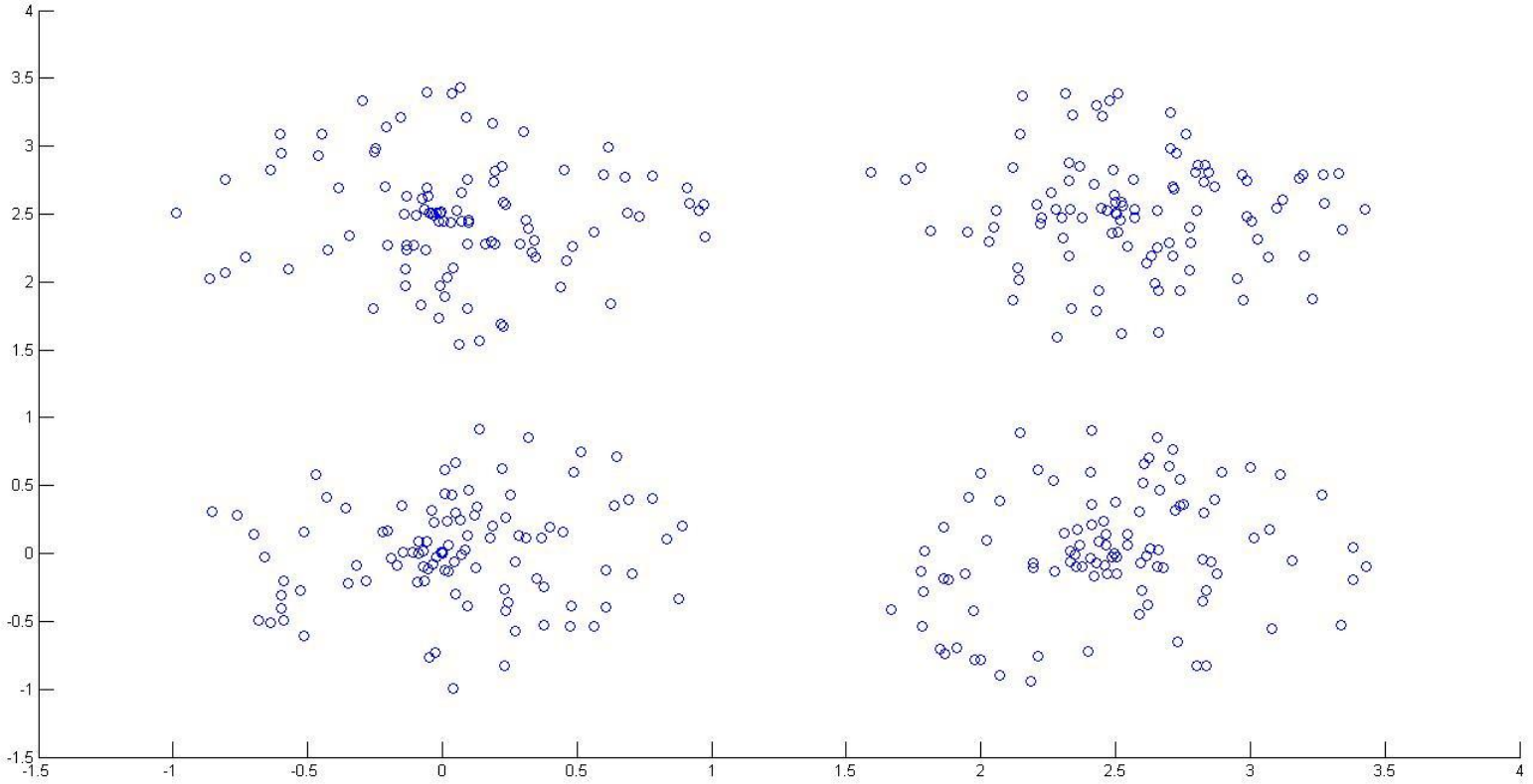
Introduzione

- Il raggruppamento di popolazioni di oggetti (unità statistiche) in base alle loro caratteristiche (variabili) è da sempre oggetto di studio:
  - classificazione delle specie animali,
  - catalogazione della flora,
  - patrimonio bibliotecario,
  - gruppi etnici, patrimonio genetico,
- Gli algoritmi di clustering si propongono di suddividere gli elementi di un insieme in gruppi omogenei di osservazioni, detti *cluster.*
- La osservazioni assegnate a ciascun cluster sono tra loro simili e risultano differenti da quelle di altri gruppi.

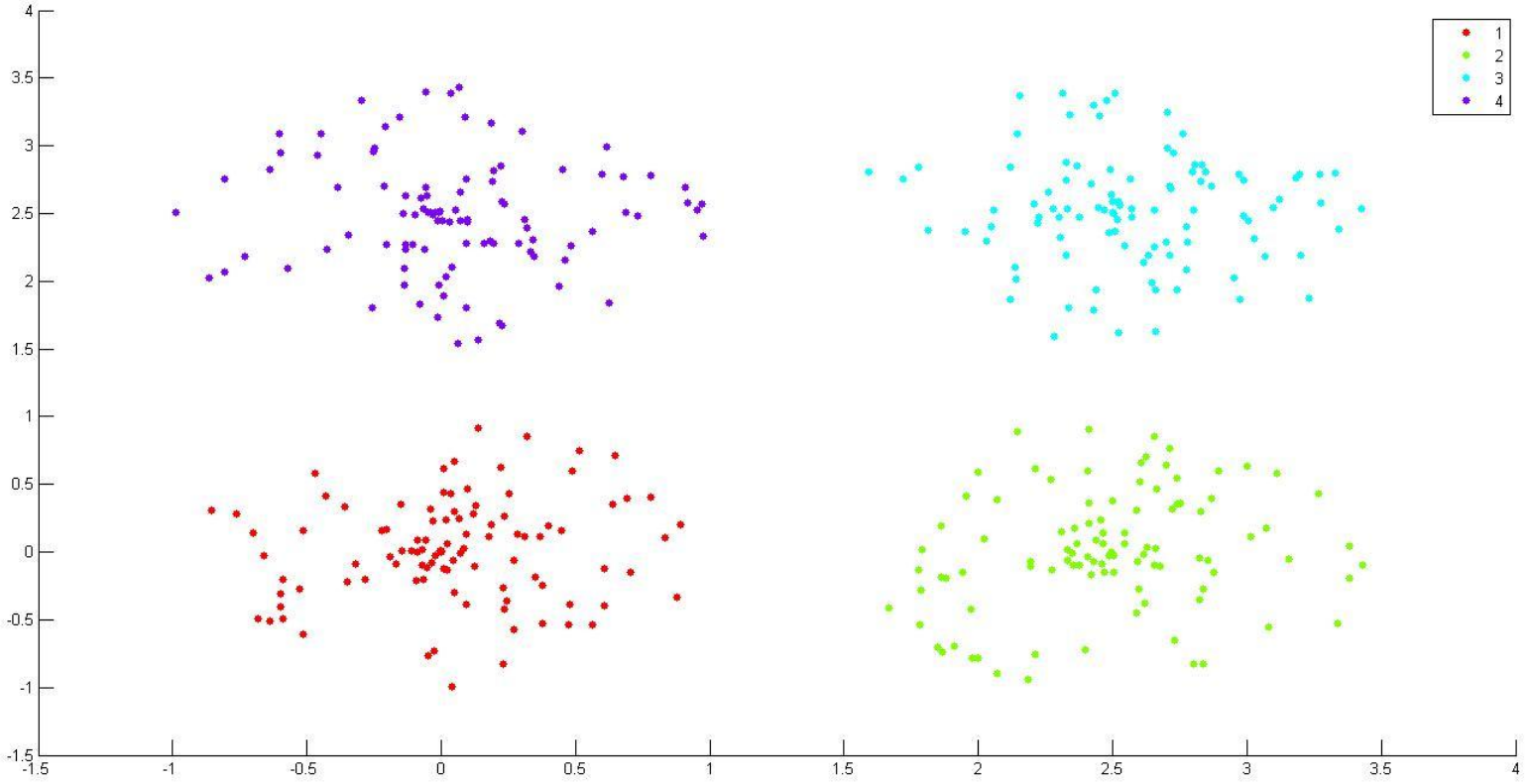
Introduzione

- Il processo di identificazione di gruppi (cluster) può avvenire con due differenti approcci:
  - **Supervisionato**: si catalogano gli oggetti in base a dei criteri prestabiliti;
  - **Non supervisionato**: l'aggregazione è meramente esplorativa e si basa sulle caratteristiche rilevate per la singola unità statistica.
- In questa lezione viene descritto l'algoritmo di clustering delle k-medie (non superivisionato) e vengono presentati alcuni indicatori di qualità.

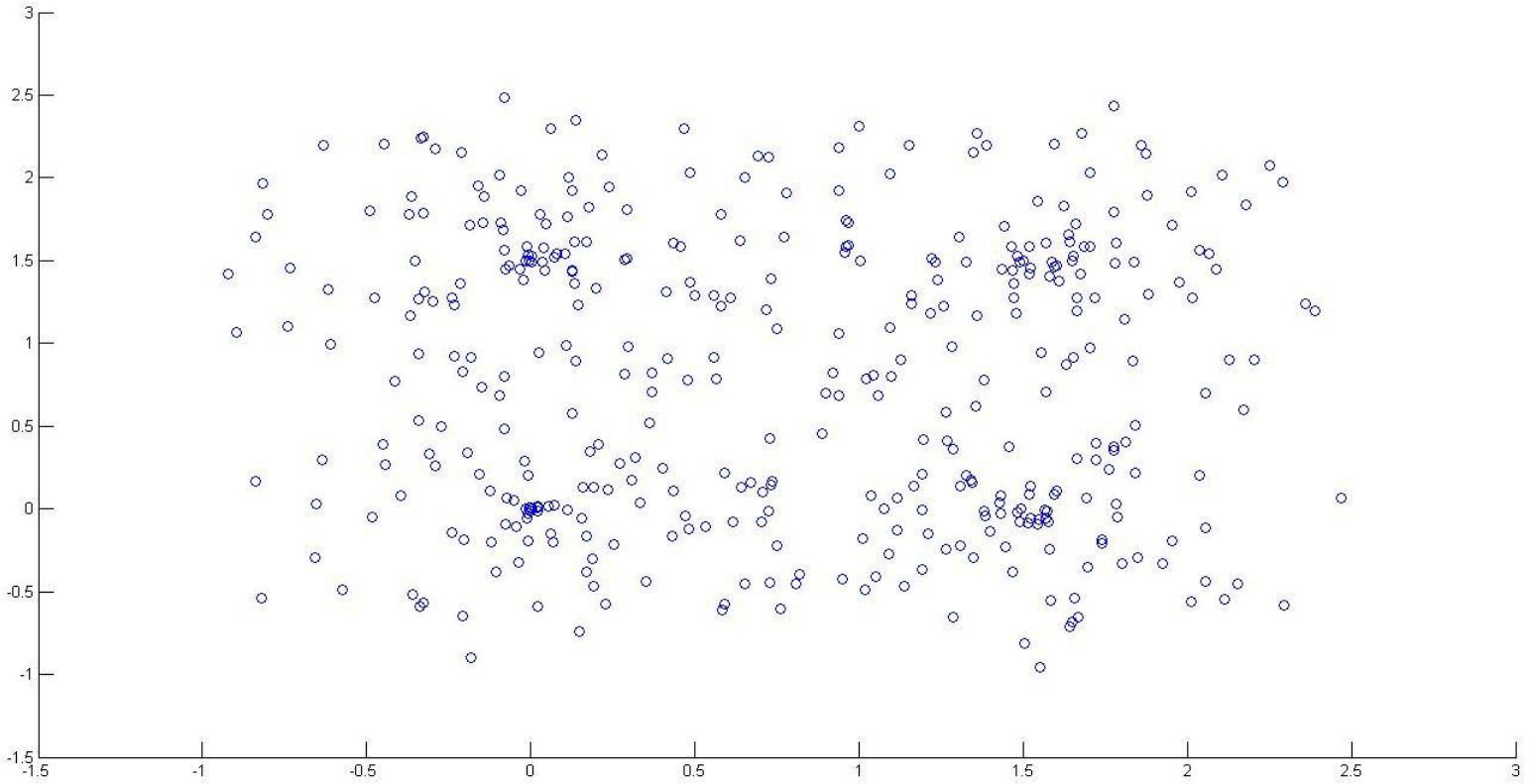
Esempio



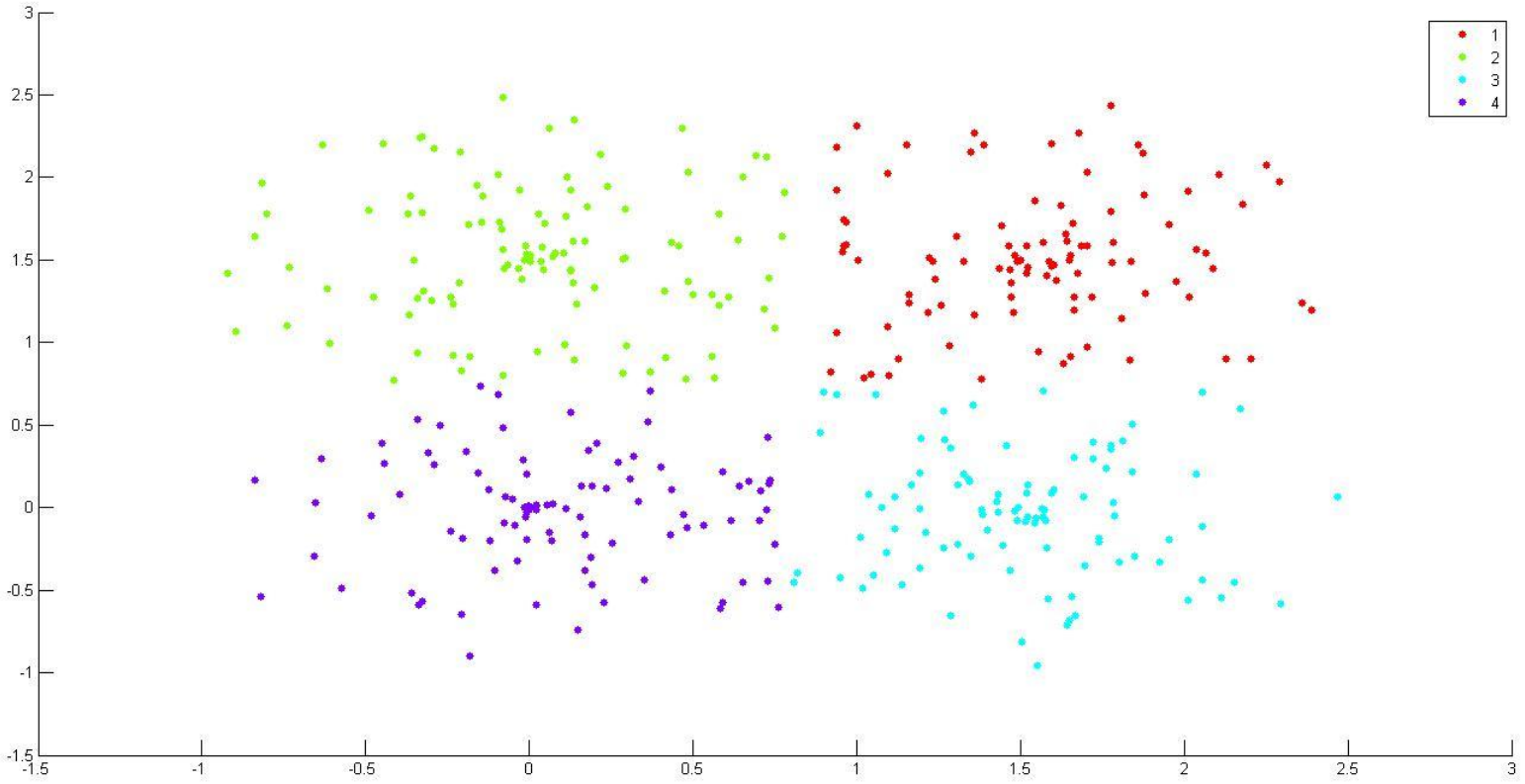
Esempio



Esempio



Esempio



Caratteristiche dei modelli di clustering

- Flessibilità: capacità grado di analizzare dati contenenti differenti tipi di attributi.
- Robustezza: stabilità rispetto a piccole perturbazioni nei dati.
- Efficienza: rapidità nella generazione dei cluster per grandi insiemi di dati e/o numerosi attributi.

Tipologie di modelli di clustering

- Tipologie:
  - **Metodi di partizione**: ricavano una partizione dei dati in K classi. Adatti a raggruppamenti di forma sferica.
  - **Metodi gerarchici**: successive suddivisione dei dati. Determinano da soli il numero di cluster
  - **Metodi basati sulla densità**: sviluppano i cluster in base al numero di elementi in ciascun intorno.
  - **Metodi a griglia**: il clustering viene effettuato discretizzando prima lo spazio.
- Si può distinguere tra metodi che assegnano o meno univocamente un elemento ad un cluster, che assegnano o meno tutti gli elementi, …

Misure

- Il clustering si basa su misure di similarità tra le osservazioni.
- Un dataset D si può rappresentare una matrice con le osservazioni sulle righe:

$$D = [d_{ik}] = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1,m-1} & d_{1m} \\ & 0 & \cdots & d_{2,m-1} & d_{2m} \\ & & \cdots & \vdots & \vdots \\ & & & 0 & d_{m-1,m} \\ & & & & 0 \end{bmatrix}$$

- La matrice simmetrica di dimensioni m x n delle distanze tra coppie di osservazioni ottenuta ponendo

$$d_{ik} = \text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \text{dist}(\mathbf{x}_k, \mathbf{x}_i), \quad i, k \in \mathcal{M},$$

con $\text{dist}(x_i, x_k)$ distanza tra le osservazioni x_i e x_k.

Misure

- Osserviamo che risulta possibile trasformare una distanza d_{ik} tra due osservazioni in una misura s_{ik} di similarità, mediante una delle seguenti relazioni

$$s_{ik} = \frac{1}{1 + d_{ik}}, \quad \text{oppure} \quad s_{ik} = \frac{d_{max} - d_{ik}}{d_{max}},$$

- dove $d_{max} = \max_{i,k} d_{ik}$ denota la massima distanza tra le osservazioni del dataset D.
- La definizione di un'appropriata nozione di distanza dipende dalla natura degli attributi che costituiscono il dataset D, che possono essere:
  - numerici, binari, categorici nominali, categorici ordinali, a composizione mista.

Misure per attributi numerici

- Se tutti gli attribuiti dal dataset sono numerici si può usare la metrica euclidea tra i vettori dlle osservazioni $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ e $x_k = (x_{k1}, x_{k2}, \ldots, x_{kn})$, definita come

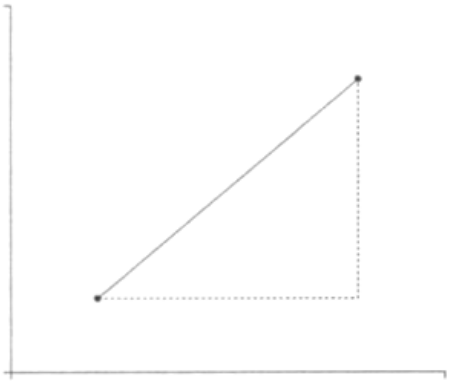
$$\begin{aligned} \text{dist}(\mathbf{x}_i, \mathbf{x}_k) &= \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{kj})^2} \\ &= \sqrt{(x_{i1} - x_{k1})^2 + (x_{i2} - x_{k2})^2 + \cdots + (x_{in} - x_{kn})^2}. \end{aligned}$$

Misure per attributi numerici

- In alternativa è possibile considerare la distanza di Manhattan

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sum_{j=1}^{n} |x_{ij} - x_{kj}| = |x_{i1} - x_{k1}| + |x_{i2} - x_{k2}| + \cdots + |x_{in} - x_{kn}|,$$

- così denominata in quanto per raggiungere un punto da un altro si percorrono i due lati di un rettangolo avente per vertici opposti i punti stessi.



Misure per attributi numerici

- Un'altra possibilità, che generalizza sia la distanza euclidea sia la distanza di Manhattan, consiste nella distanza di Minkowski:

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt[1/q]{\sum_{j=1}^{n} |x_{ij} - x_{kj}|^q} = \sqrt[1/q]{|x_{i1} - x_{k1}|^q + |x_{i2} - x_{k2}|^q + \ldots, |x_{in} - x_{kn}|^q},$$

- dove q è un numero intero positivo assegnato.
- La distanza di Minkowski si riduce alla distanza di Manhattan per q = 1, e alla distanza di Euclide per q = 2.

Misure per attributi numerici

- Un‘ultima generalizzazione della distanza euclidea si ottiene mediante la distanza di Mahalanobis, difinita come

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \sqrt{(\mathbf{x}_i - \mathbf{x}_k)\mathbf{V}_{ik}^{-1}(\mathbf{x}_i - \mathbf{x}_k)'},$$

- dove $\mathbf{V}_{ik}^{-1}$ rappresenta l'inverso della matrice di covarianza della coppia di osservazioni x_i e x_k.
- Se le osservazioni x_i e x_k sono indipendenti, e la matrice di covarianza si riduce pertanto alla matrice identità, la distanza di Mahalanobis coincide con la distanza euclidea.

Misure per attributi numerici

- Le espressioni descritte per la distanza tra due osservazioni risentono di eventuali attributi che hanno valori assoluti grandi rispetto agli altri.
- In situazioni estreme, un solo attributo dominante potrebbe condizionare la formazione dei raggruppamenti da parte degli algoritmi di clustering.
- Esistono alcuno accorgimenti per evitare un simile sbilanciamento nella valutazione delle distanze tra le osservazioni:
  - standardizzazione dei valori degli attributi numerici, in modo da ottenere nuovi valori che si collocano nell'intervallo [-1,1].

Misure per attributi binari

- Se un attributo di $x_j = (x_{1j}, x_{2j}, \ldots, x_{mj})$ è binario, allora assume soltanto uno dei due valori 0 o 1.
- Anche se è possibile calcolare $x_{ij} - x_{kj}$, questa quantità non rappresenta una distanza significativa, come per le metriche che abbiamo definito per gli attributi numerici.
  - i valori 0 e 1 possono essere puramente convenzionali, e potrebbero essere scambiati di significato tra loro.

Misure per attributi binari

- Supponiamo che nel dataset D tutti gli n attributi presenti siano binari.
- Per procedere nella definizione di una metrica occorre fare riferimento alla tabella

		osservazione x_k		
		0	1	totale
osservazione x_i	0	p	q	$p+q$
	1	u	v	$u+v$
	totale	$p+u$	$q+v$	n

Tabella 12.1 Tabella di contingenza per un attributo binario.

- Il valore p rappresenta il numero di attributi binari in corrispondenza dei quali entrambe le osservazioni xi e xk assumono valore 0. Analogamente per q, u e v.
  - Vale la relazione n = p + q + u + v.

Misure per attributi binari

- Gli attributi binari possono essere:
  - **Simmetrici**: la presenza del valore 0 è interessante quanto la presenza del valore 1.
  - **Asimmetrici**: siamo invece interessati in modo prevalente alla presenza del valore 1, che può manifestarsi in una piccola percentuale delle osservazioni.
- Se tutti gli n attributi binari sono di natura simmetrica possiamo definire il grado di similarità tra le osservazioni con il coefficiente delle corrispondenze:

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q+u}{p+q+u+v} = \frac{q+u}{n}.$$

Misure per attributi binari

- Supponiamo invece che tutti gli n attributi siano binari e asimmetrici.
- Per una coppia di attributi asimmetrici risulta molto più interessante l'abbinamento dei positivi.
- Per le variabili binarie asimmetriche si ricorre quindi al coeffciente di Jaccard, definito come:

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{q + u}{p + q + u}.$$

Misure per attributi categorici nominali

- Un attributo categorico nominale si può interpretare come un attributo binario simmetrico, con un numero di valori assunti maggiore di 2.
- Quindi, anche per gli attributi nominali si può estendere il coefficiente delle corrispondenze, definito per variabili binarie simmetriche, mediante la nuova espressione

$$\text{dist}(\mathbf{x}_i, \mathbf{x}_k) = \frac{n - f}{n},$$

- dove f indica il numero di attributi per i quali le osservazioni x_i e x_k assumono lo stesso valore nominale.

Metodi per attributi categorici ordinali

- Gli attributi categorici ordinali possono essere collocati su una scala di ordinamento naturale, con valori numerici arbitrari.
- Richiedono pertanto una standardizzazione preliminare per poter essere adattati alle metriche definite per gli attributi numerici.

Metodi per attributi categorici ordinali

- Supponiamo di rappresentare i valori di ciascun attributo categorico ordinale mediante la sua posizione nell'ordinamento naturale.
- Se ad es. la variabile corrisponde al grado di istruzione, e può assumere i livelli {elementari, medie, diploma, laurea}, si fa corrispondere il livello {elementari} al valore numerico 1, il livello {medie} al valore 2, e cosi via.
- Indichiamo con Hj = {1, 2 ,...Hj} i valori ordinati associati all'attributo ordinale aj.

Metodi per attributi categorici ordinali

- Per standardizzare i valori assunti dall'attributo aj nell'intervallo [0,1] si ricorre alla trasformazione

$$x'_{ij} = \frac{x_{ij} - 1}{H_j - 1}.$$

- Dopo avere effettuato la trasformazione indicata per tutte le variabili ordinali, è possibile ricorrere alle misure di distanza per gli attributi numerici.

Metodi di partizione

- I metodi di partizione suddividono un dataset D in K sottogruppi non vuoti C = {C1, C2, ..., CK}, dove K ≤ m.
- In genere il numero K di cluster viene assegnato in ingresso agli algoritmi di partizione.
- I gruppi generati risultano di solito esaustivi ed esclusivi, nel senso che ogni osservazione appartiene a uno e un solo cluster.
- I metodi di partizione partono con un'assegnazione iniziale delle osservazioni ai K cluster.
- In seguito applicano iterativamente una tecnica di riallocazione delle osservazioni per assegnare alcune osservazioni a un diverso cluster, in modo da accrescere la qualità complessiva della suddivisione.

Metodi di partizione

- Le diverse misure di qualità utilizzate tendono a esprimere il grado di omogeneità delle osservazioni appartenenti a un medesimo cluster e la loro eterogeneità rispetto a osservazioni collocate in altri raggruppamenti.
- Gli algoritmi si arrestano quando nel corso di un'iterazione non si verifica alcuna riallocazione, e pertanto la suddivisione appare stabile rispetto al criterio di valutazione utilizzato.
- I metodi di partizione hanno quindi natura euristica e operano a ogni passo la scelta che appare localmente più vantaggiosa.

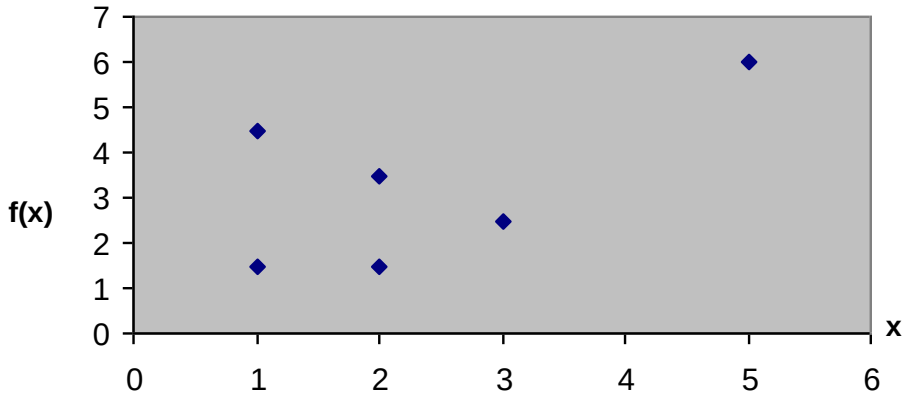
Metodi di partizione

- Procedendo in questo modo non si ha la garanzia di raggiungere una suddivisione che sia globalmente ottimale.
- Il metodo delle k-medie (k-means) è l'algoritmo di partizione più noto.
- Si tratta di un metodo di clustering abbastanza efficiente per determinare raggruppamenti di forma sferica.

Algoritmo delle k-medie

- Tecnica semplice ma efficace:
  1. Scegliere un valore di k, il numero di cluster da generare.
  2. Scegliere in modo casuale k osservazioni nel dataset.
     - Questi saranno i centri dei cluster.
  3. Collocare ogni altra osservazione nel cluster con il centro più vicino a essa.
  4. Utilizzare le osservazioni in ogni cluster per calcolarne il nuovo centro.
  5. Se i cluster non si sono modificati allora terminare, altrimenti ripetere il ciclo.
- Occorre definire un concetto di distanza (vicinanza).
- La distanza più comune fra osservazioni numeriche è quella euclidea della somma dei quadrati degli scarti.

Osservazione	X	Y
1	1.0	1.5
2	1.0	4.5
3	2.0	1.5
4	2.0	3.5
5	3.0	2.5
6	5.0	6.0



La distanza euclidea fra il punto (x_1, y_1) e il punto (x_2, y_2) è definita da:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Sia *k* = 2 (vogliamo due cluster) e i due centri a caso sono le osservazioni 1 e 3.
Quindi i centri sono A = (1.0, 1.5) e B = (2.0, 1.5).
L'osservazione 5 ha distanza da A uguale a

$$\sqrt{(3.0-1.0)^2 + (2.5-1.5)^2} = \sqrt{2^2 + 1^2} = \sqrt{5} = 2.24$$

La distanza da B invece è

$$\sqrt{(3.0-2.0)^2 + (2.5-1.5)^2} = \sqrt{1^2 + 1^2} = \sqrt{2} = 1.41$$

Quindi l'osservazione 5 va nel secondo cluster, quello che ha centro B.

L’osservazione 4 ha distanza 2.24 da A e 2.00 da B, quindi è nel 2° cluster.
L’osservazione 6 ha distanza 6.02 da A e 5.41 da B, quindi è nel 2° cluster.
L’osservazione 2 ha distanza 3.00 da A e 3.16 da B, quindi è nel 1° cluster.

I due cluster dopo il primo passo risultano quindi essere

{1, 2} e {3, 4, 5, 6}.

Il nuovo centro del primo cluster è il nuovo punto A di coordinate (1.0, 3.0), calcolate come medie delle coordinate dei due punti nel primo cluster:

nuova x di A = (1.0 + 1.0) / 2 = 1.0 e

nuova y di A = (1.5 + 4.5) / 2 = 3.0

Il nuovo centro B del secondo cluster ha coordinate (3.0, 3.375), calcolate come

nuova x di B = (2.0 + 2.0 + 3.0 + 5.0) / 4 = 3.0

nuova y di B = (1.5 + 3.5 + 2.5 + 6.0) / 4 = 3.375

I centri sono cambiati. Quindi ripetiamo il ciclo.

Nota: i nuovi centri non sono più osservazioni, ma punti inesistenti nel dataset.

Troviamo che l’osservazione 3 è più vicina al nuovo A (distanza 1.80) che al nuovo B (distanza 2.125) quindi si sposta dal secondo al primo cluster.
Anche le osservazioni 1 e 2 sono più vicine ad A che a B, quindi restano nel primo cluster.
Invece le osservazioni 4, 5 e 6 sono più vicine a B e restano nel secondo cluster.
Dopo il secondo ciclo dell’algoritmo i due cluster sono

{1, 2, 3} e {4, 5, 6}

E si continua finché i cluster si stabilizzano.

Osservazioni

- Il risultato dipende dalla scelta iniziale dei centri.
- Per questo motivo si esegue l'algoritmo varie volte con diverse scelte iniziali e si cerca un risultato "buono", anche se non proprio ottimo
  - (non si possono provare tutte le combinazioni perché numerosissime).
- Un clustering è buono se si ha una piccola varianza nei cluster e grande varianza fra i cluster.
- La varianza entro un cluster è la somma dei quadrati delle distanze dei punti dal centro del cluster.
- La varianza fra cluster è la varianza dei centri dei cluster rispetto al centro dei centri.

Valutazione dei modelli di clustering

- i Indichiamo con C = {C_1 , C_2, ... , C_k} l'insieme di K cluster generati da un algoritmo di clustering.
- Un indicatore di omogeneità di ciascun cluster è dato da:

$$\text{coes}(C_h) = \sum_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_h}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k).$$

detta ***coesione*** del cluster.

- Si può quindi definire la ***coesione della partizione***:

$$\text{coes}(\mathcal{C}) = \sum_{C_h \in \mathcal{C}} \text{coes}(C_h).$$

Valutazione dei modelli di clustering

- Un indicatore della disomogeneità tra una coppia di cluster è dato da:

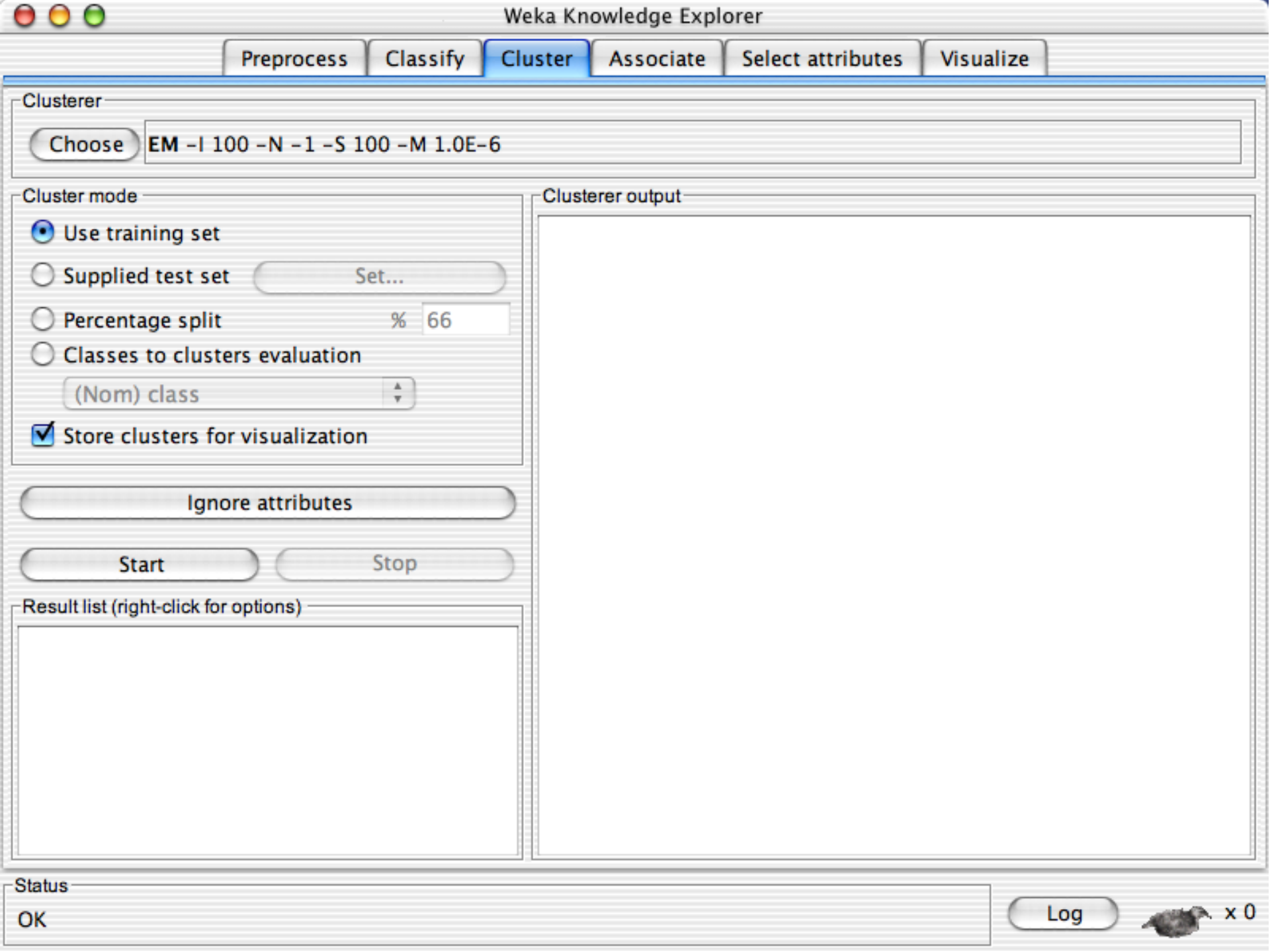
$$\text{sep}(C_h, C_f) = \sum_{\substack{\mathbf{x}_i \in C_h \\ \mathbf{x}_k \in C_f}} \text{dist}(\mathbf{x}_i, \mathbf{x}_k).$$

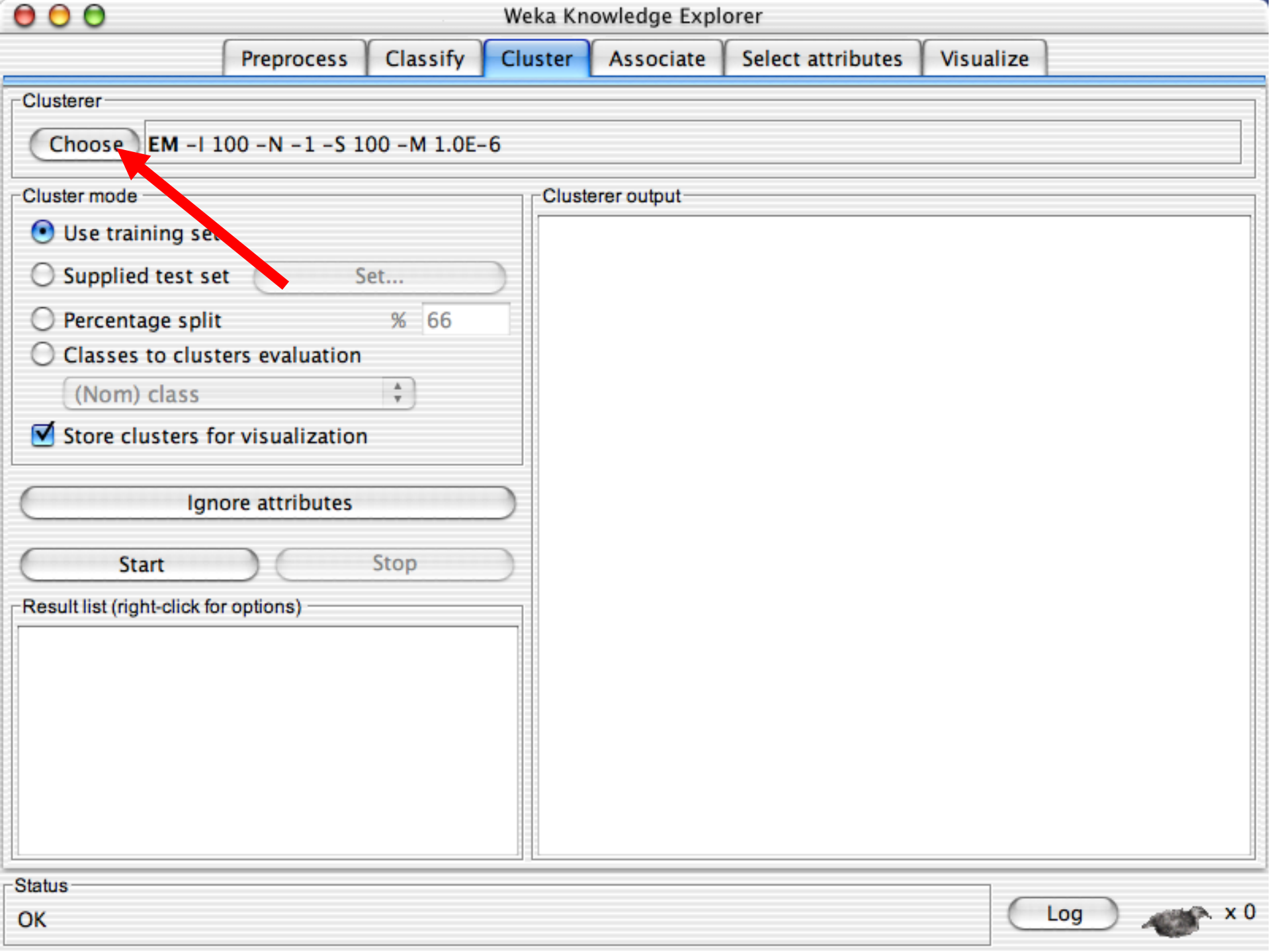
- Anche in questo caso, la separazione complessiva della partizione è data da:

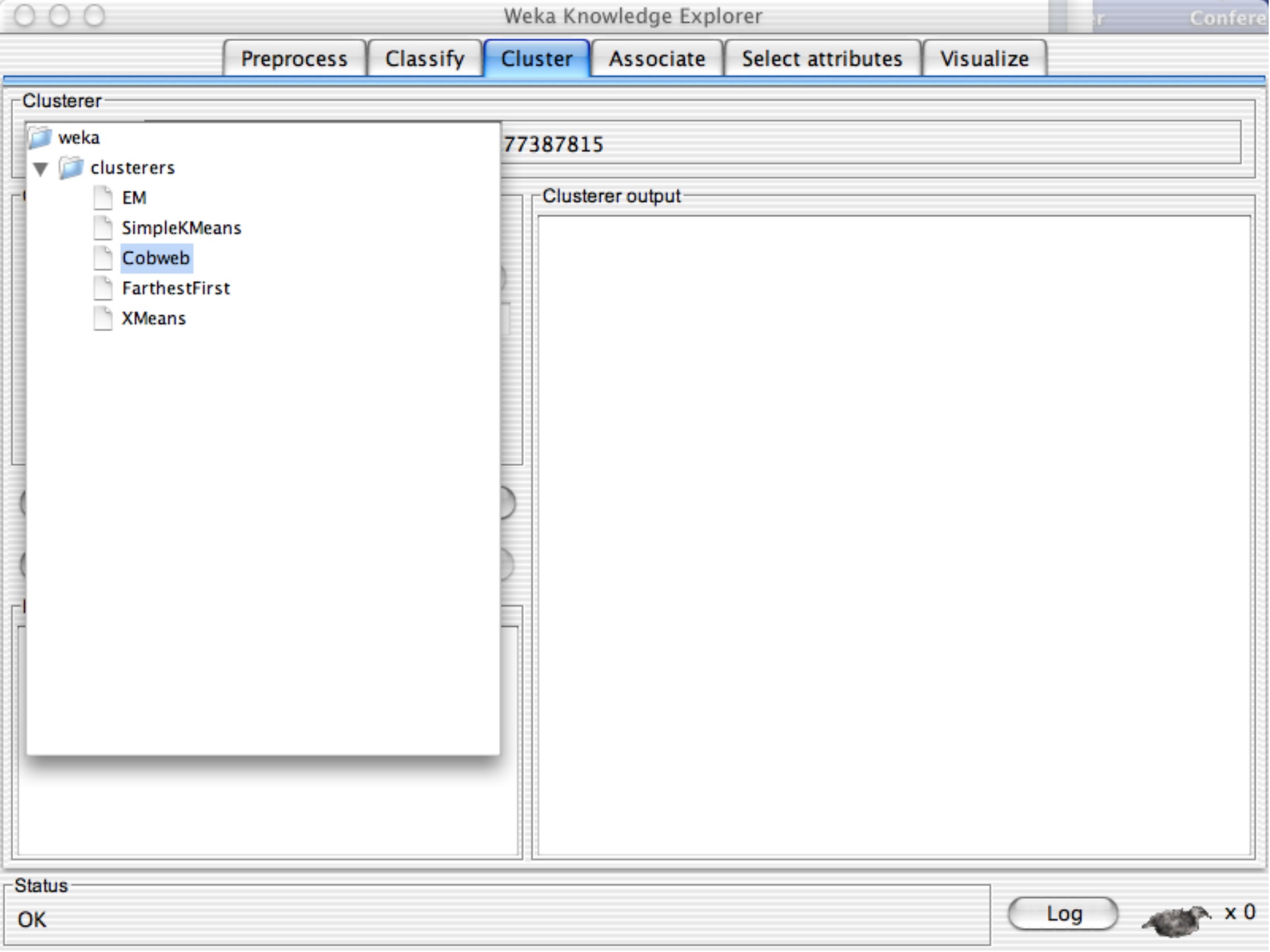
$$\text{sep}(\mathcal{C}) = \sum_{\substack{C_h \in \mathcal{C} \\ C_f \in \mathcal{C}}} \text{sep}(C_h, C_f).$$

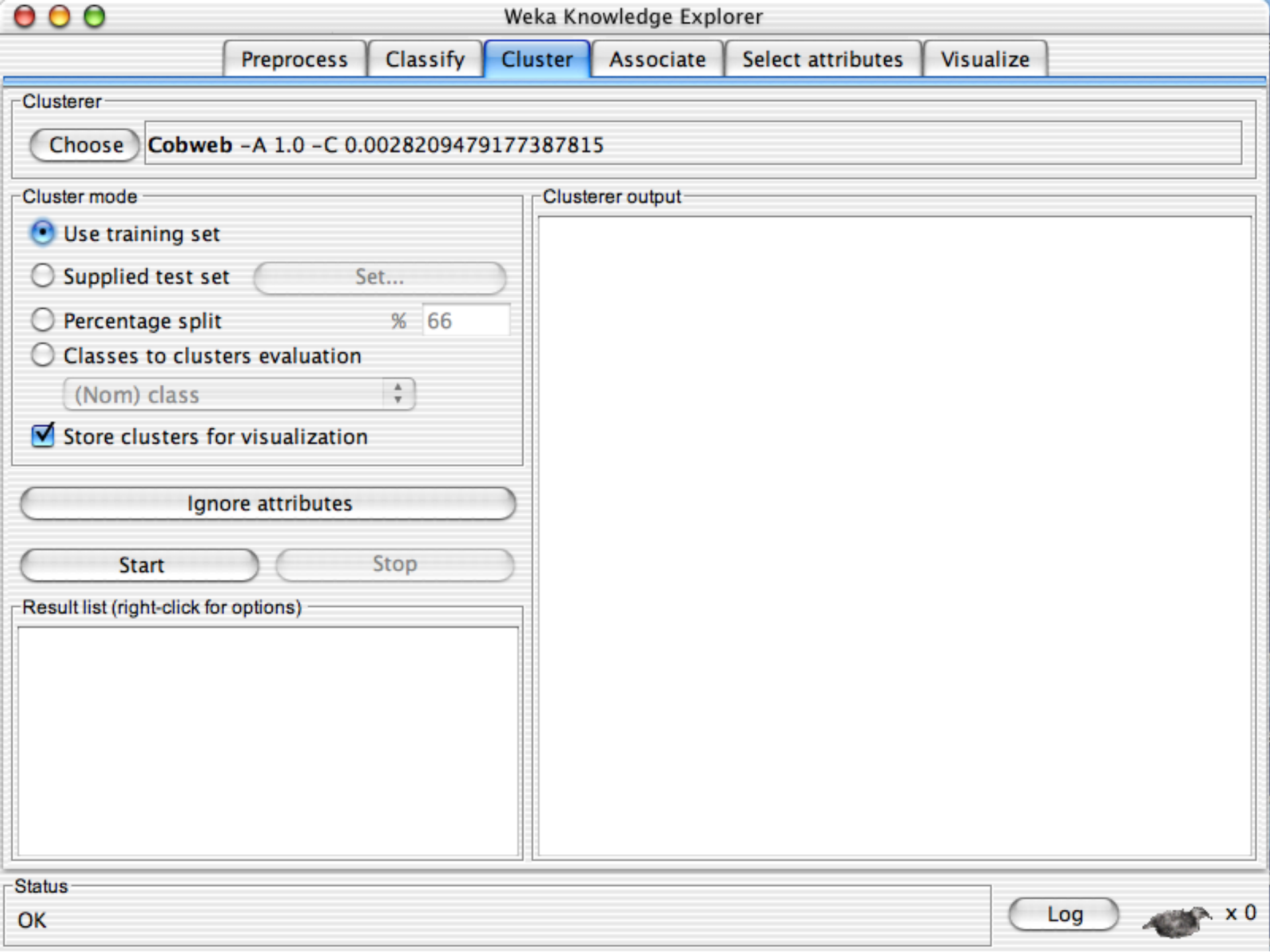
Explorer: clustering data in Weka

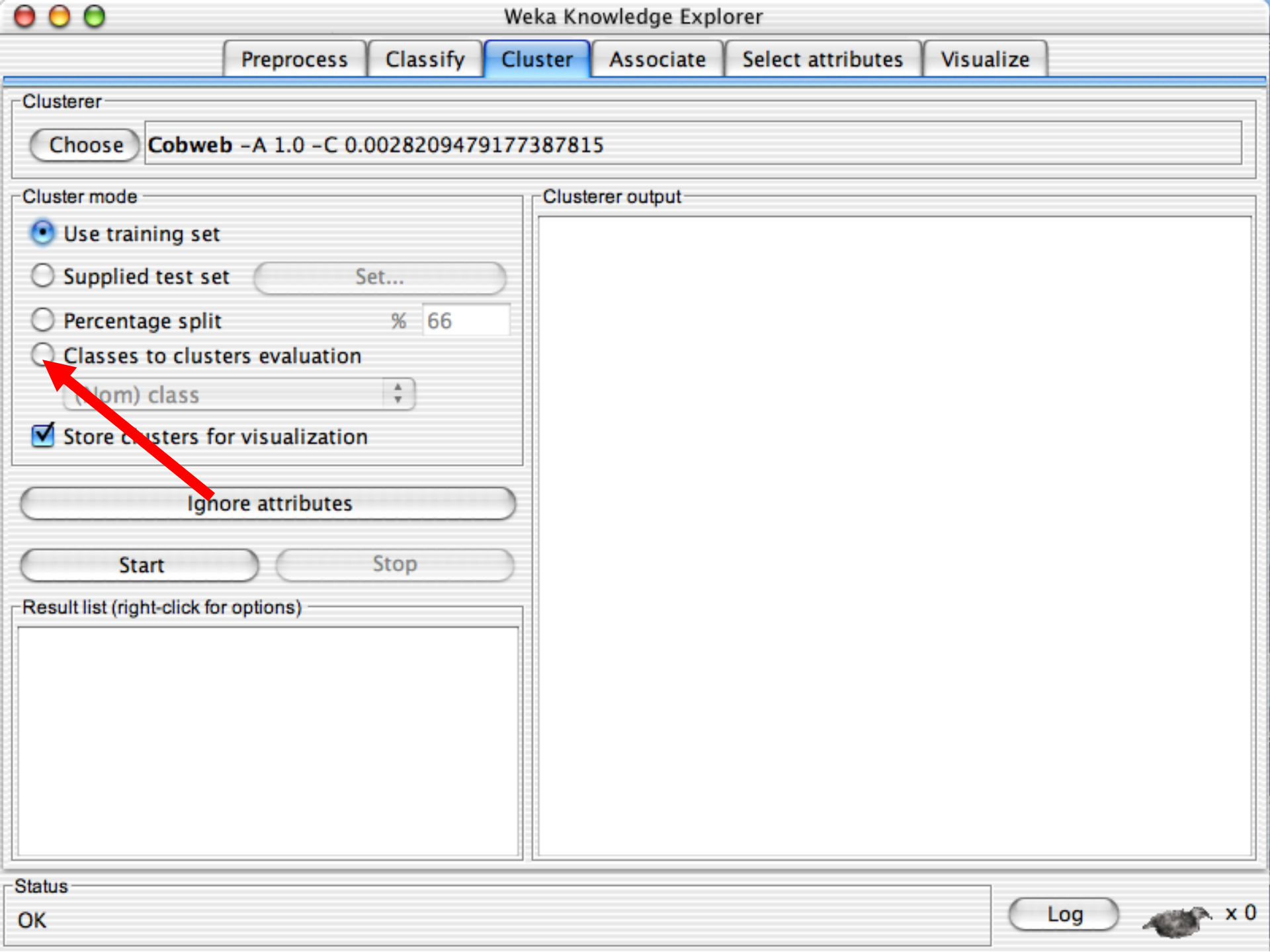
- WEKA contains “clusterers” for finding groups of similar instances in a dataset
- Implemented schemes are:
  - *k*-Means, EM, Cobweb, *X*-means, FarthestFirst
- Clusters can be visualized and compared to “true” clusters (if given)
- Evaluation based on loglikelihood if clustering scheme produces a probability distribution

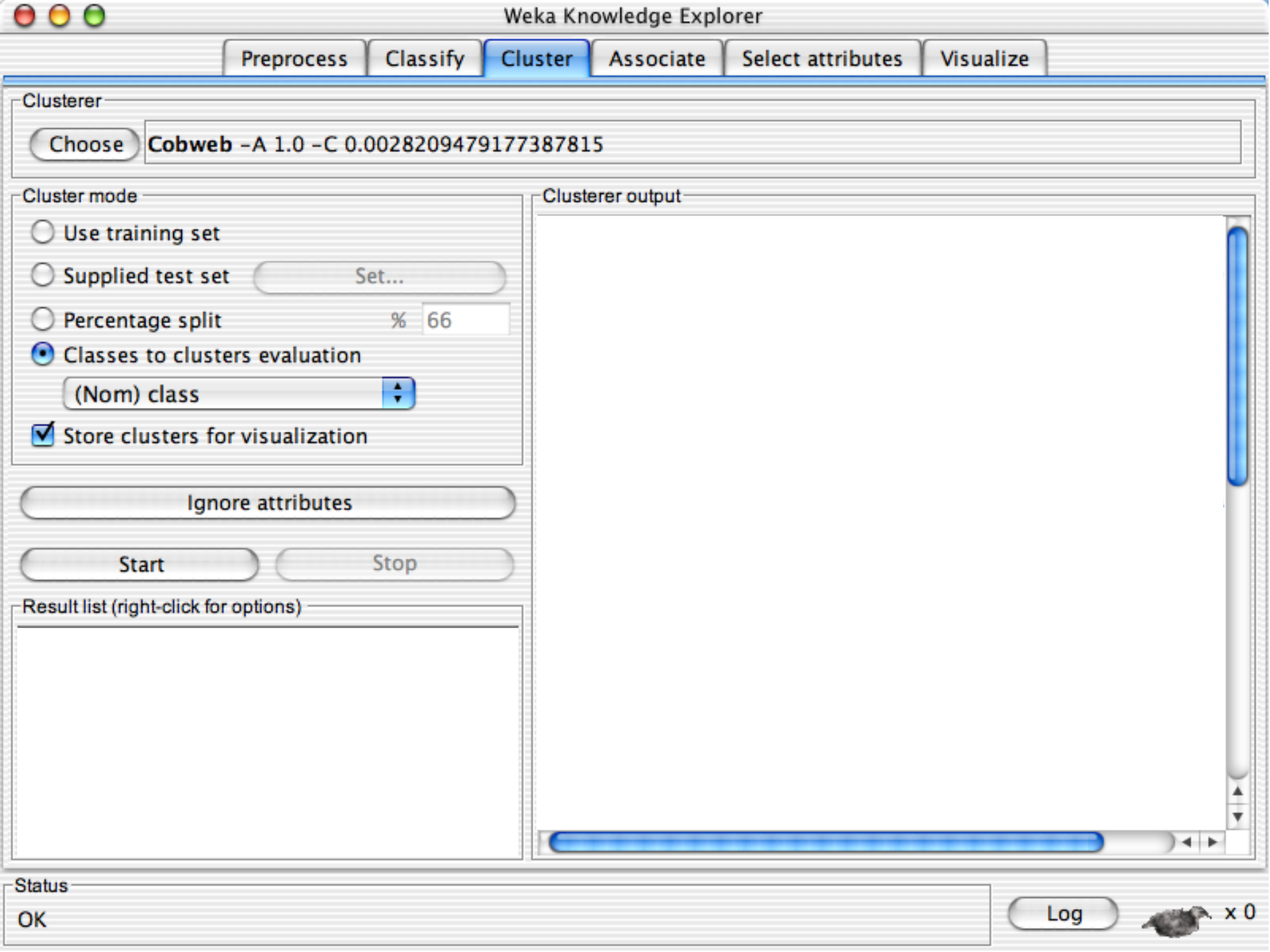

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -S 100 -M 1.0E-6
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0

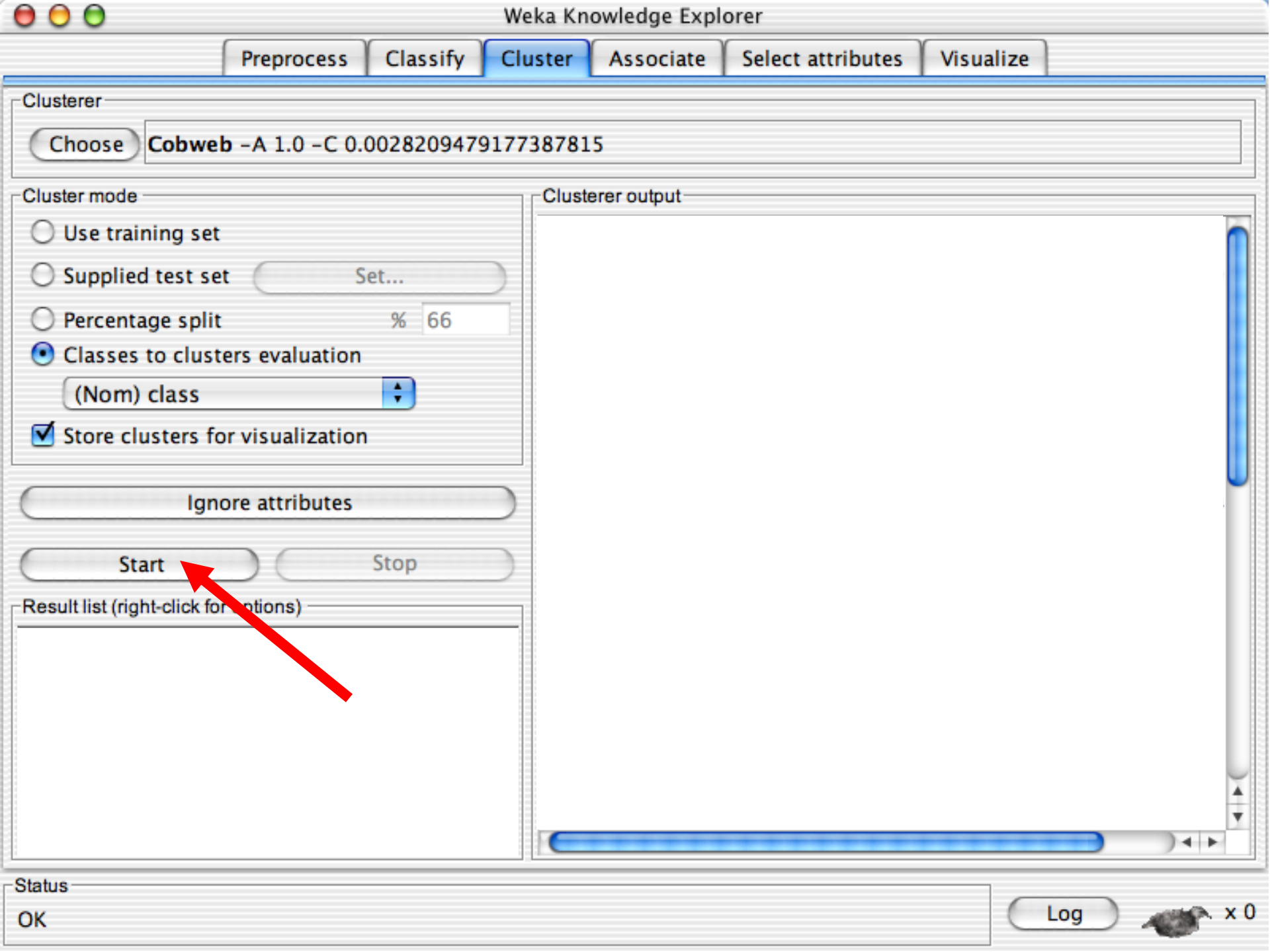

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
EM -I 100 -N -1 -S 100 -M 1.0E-6
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0

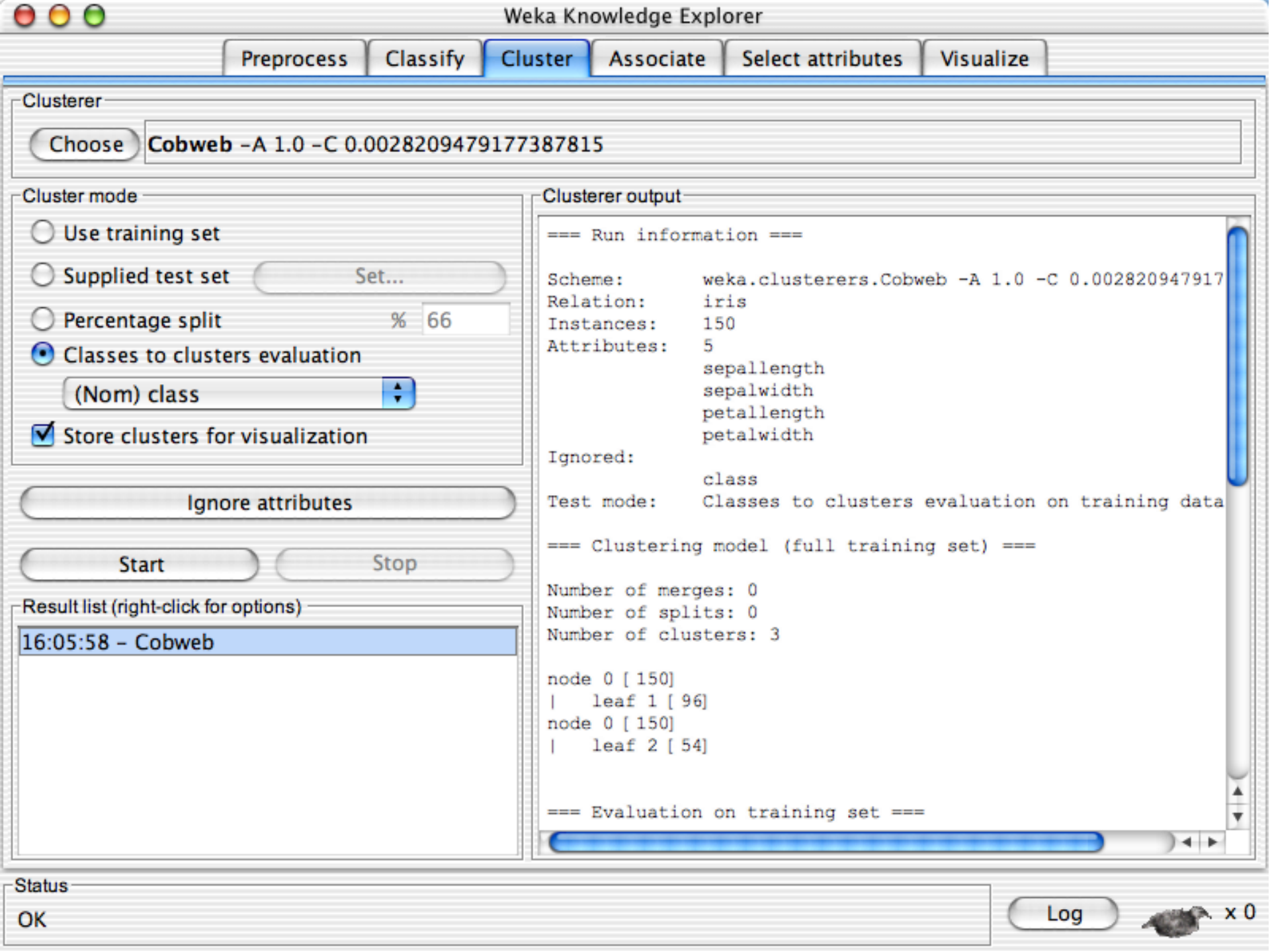

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
77387815
weka
clusterers
EM
SimpleKMeans
Cobweb
FarthestFirst
XMeans
Clusterer output
Status
OK
Log
x 0

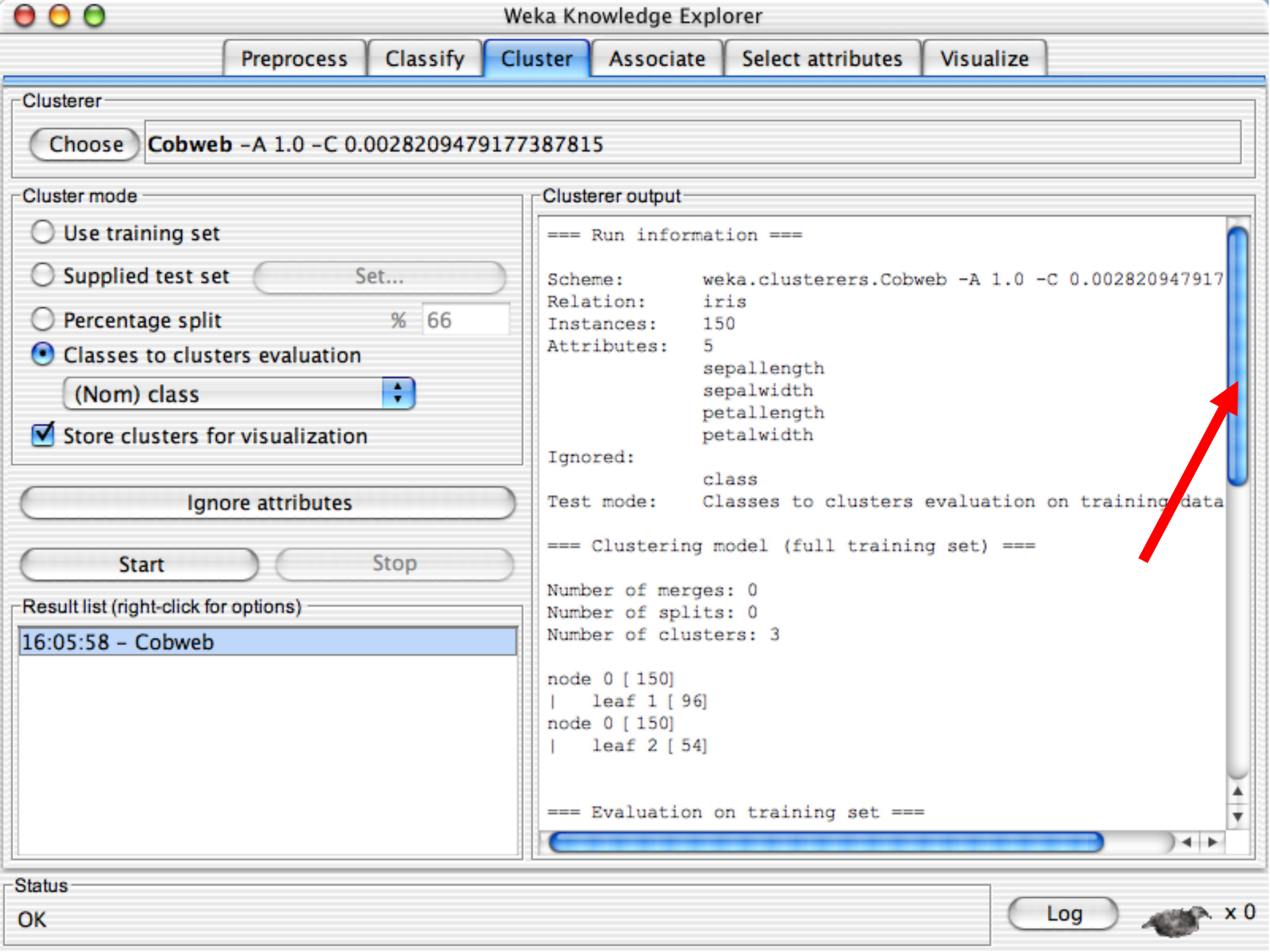

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Clusterer output
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:05:58 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.002820947917
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 0
Number of splits: 0
Number of clusters: 3
node 0 [150]
| leaf 1 [96]
node 0 [150]
| leaf 2 [54]
=== Evaluation on training set ===
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:05:58 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.002820947917
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 0
Number of splits: 0
Number of clusters: 3
node 0 [150]
| leaf 1 [96]
node 0 [150]
| leaf 2 [54]
=== Evaluation on training set ===
Status
OK
Log
x 0

Weka Knowledge Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose **Cobweb** -A 1.0 -C 0.0028209479177387815

Cluster mode

- Use training set
- Supplied test set — Set...
- Percentage split — % 66
- Classes to clusters evaluation
  - (Nom) class
- Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

16:05:58 - Cobweb

Clusterer output

```
Number of clusters: 3

node 0 [150]
|   leaf 1 [96]
node 0 [150]
|   leaf 2 [54]

Clustered Instances

1       100 ( 67%)
2        50 ( 33%)


Class attribute: class
Classes to Clusters:

  1  2  <-- assigned to cluster
  0 50 | Iris-setosa
 50  0 | Iris-versicolor
 50  0 | Iris-virginica

Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       50.0     33.3333 %
```

Status

OK

Log

x 0

Weka Knowledge Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Clusterer

Choose **Cobweb** -A 1.0 -C 0.0028209479177387815

Cluster mode

- Use training set
- Supplied test set — Set...
- Percentage split — % 66
- Classes to clusters evaluation
  - (Nom) class
- Store clusters for visualization

Ignore attributes

Start | Stop

Result list (right-click for options)

16:05:58 - Cobweb

Clusterer output

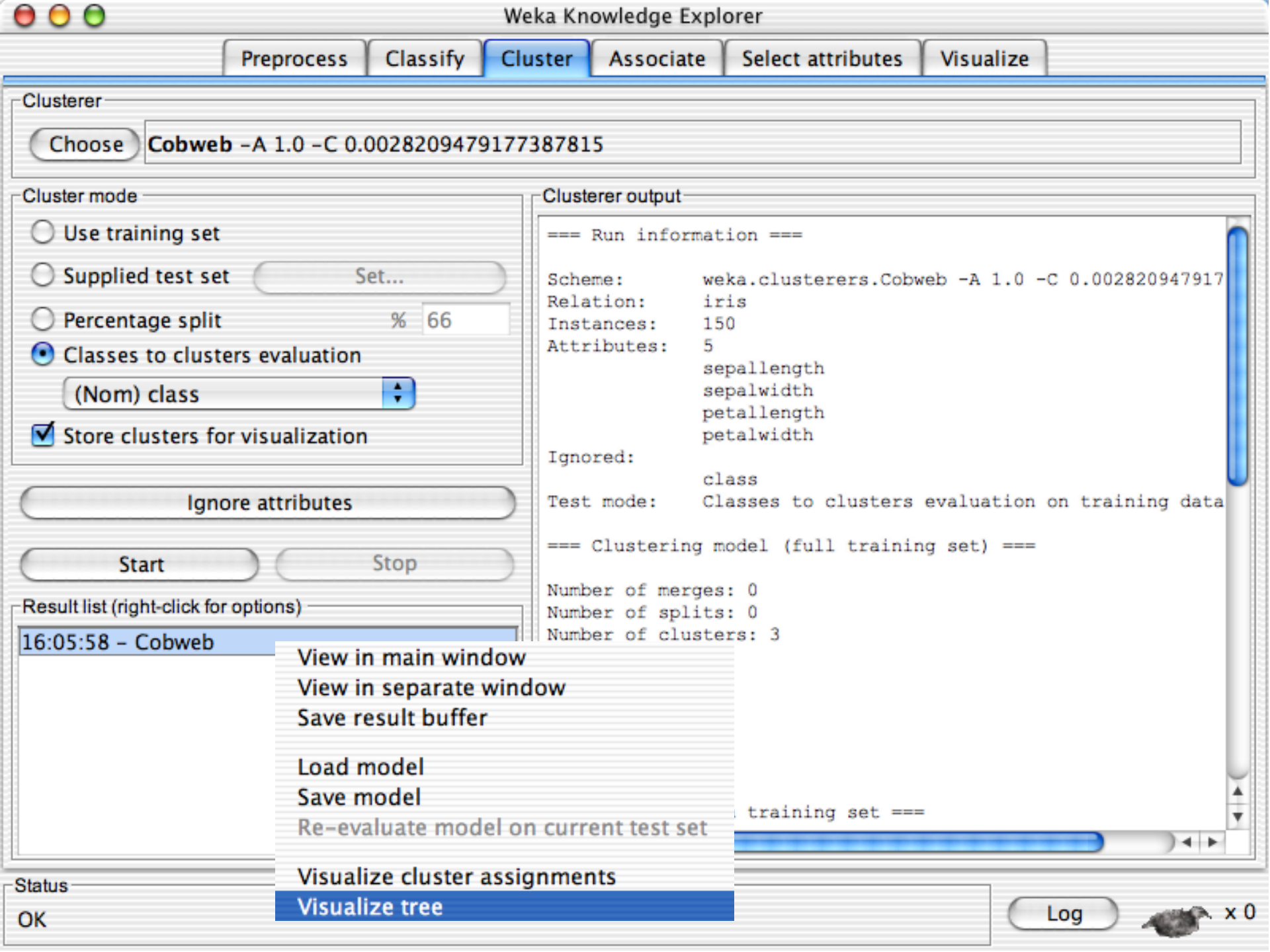
```
Number of clusters: 3

node 0 [150]
|   leaf 1 [96]
node 0 [150]
|   leaf 2 [54]

Clustered Instances

1       100 ( 67%)
2        50 ( 33%)


Class attribute: class
Classes to Clusters:

  1  2  <-- assigned to cluster
  0 50 | Iris-setosa
 50  0 | Iris-versicolor
 50  0 | Iris-virginica

Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-setosa

Incorrectly clustered instances :       50.0     33.3333 %
```

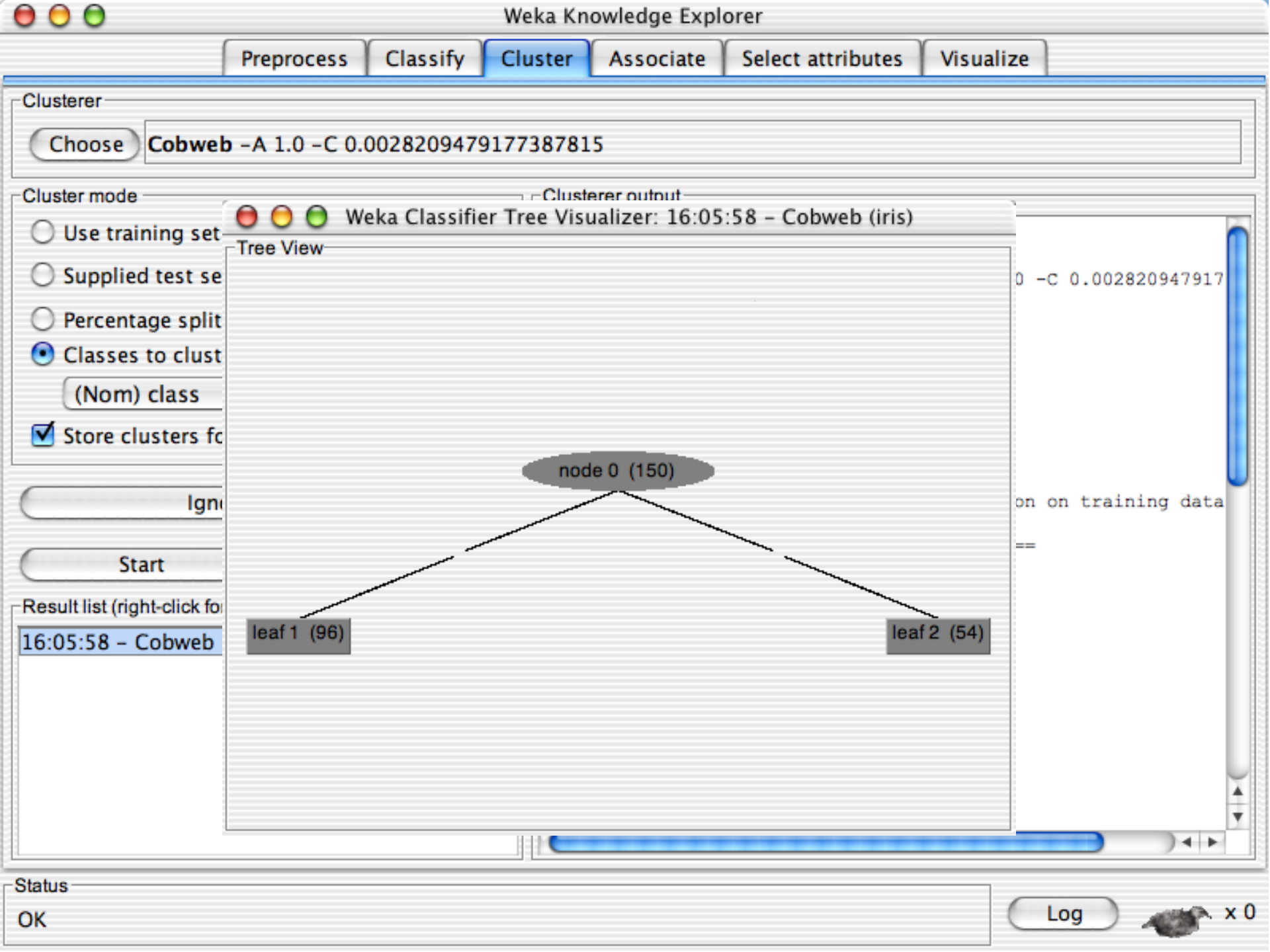
Status

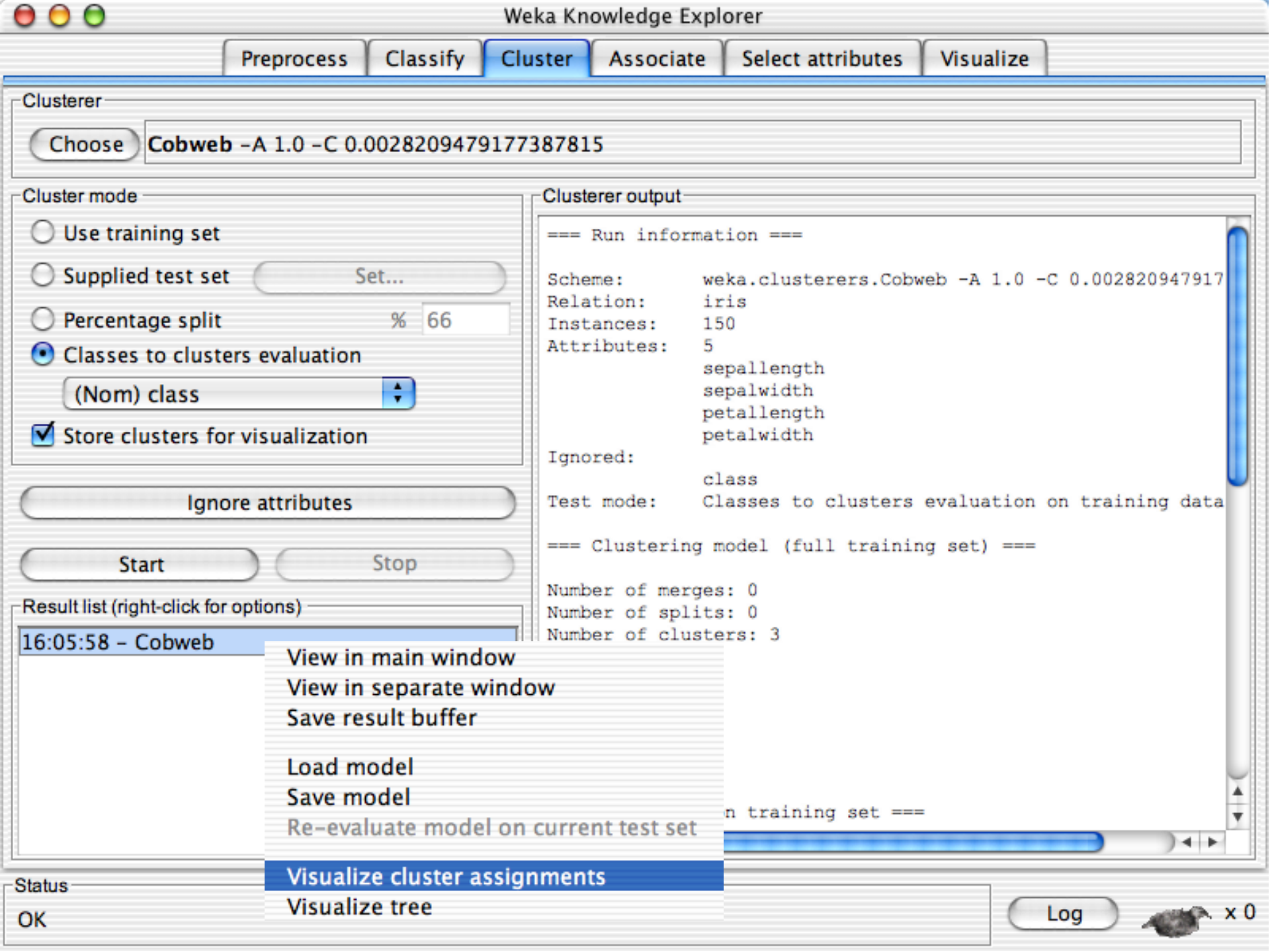
OK

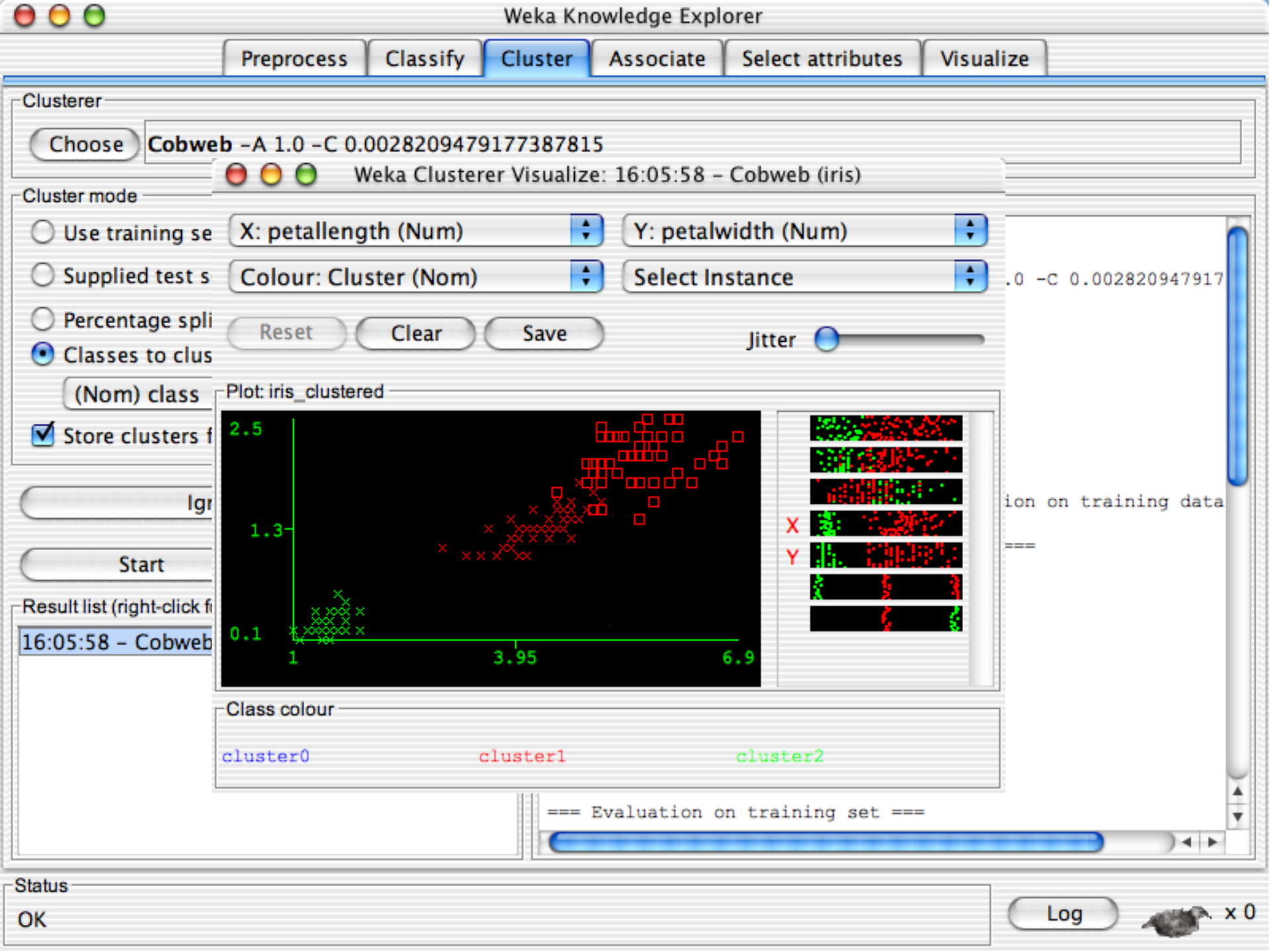
Log

x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:05:58 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.002820947917
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 0
Number of splits: 0
Number of clusters: 3
training set ===
View in main window
View in separate window
Save result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize cluster assignments
Visualize tree
Status
OK
Log
x 0

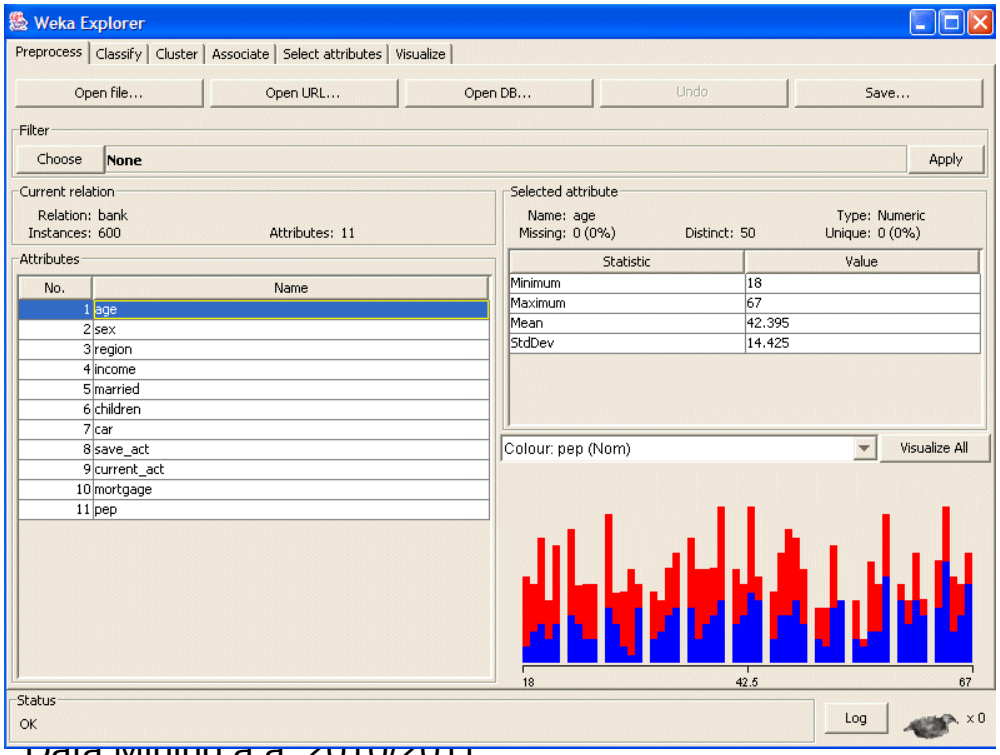

Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test se
Percentage split
Classes to clust
(Nom) class
Store clusters fo
Ignore
Start
Result list (right-click for
16:05:58 - Cobweb
Clusterer output
0 -C 0.002820947917
on on training data
==
Weka Classifier Tree Visualizer: 16:05:58 - Cobweb (iris)
Tree View
node 0 (150)
leaf 1 (96)
leaf 2 (54)
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
% 66
Classes to clusters evaluation
(Nom) class
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:05:58 - Cobweb
Clusterer output
=== Run information ===
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.002820947917
Relation: iris
Instances: 150
Attributes: 5
sepallength
sepalwidth
petallength
petalwidth
Ignored:
class
Test mode: Classes to clusters evaluation on training data
=== Clustering model (full training set) ===
Number of merges: 0
Number of splits: 0
Number of clusters: 3
n training set ===
View in main window
View in separate window
Save result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize cluster assignments
Visualize tree
Status
OK
Log
x 0


Weka Knowledge Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
Cobweb -A 1.0 -C 0.0028209479177387815
Weka Clusterer Visualize: 16:05:58 - Cobweb (iris)
Cluster mode
Use training se
Supplied test s
Percentage spli
Classes to clus
(Nom) class
Store clusters f
X: petallength (Num)
Y: petalwidth (Num)
Colour: Cluster (Nom)
Select Instance
Reset
Clear
Save
Jitter
Plot: iris_clustered
2.5
1.3
0.1
1
3.95
6.9
X
Y
Class colour
cluster0
cluster1
cluster2
Start
Result list (right-click f
16:05:58 - Cobweb
.0 -C 0.002820947917
ion on training data
===
=== Evaluation on training set ===
Status
OK
Log
x 0

Esempio

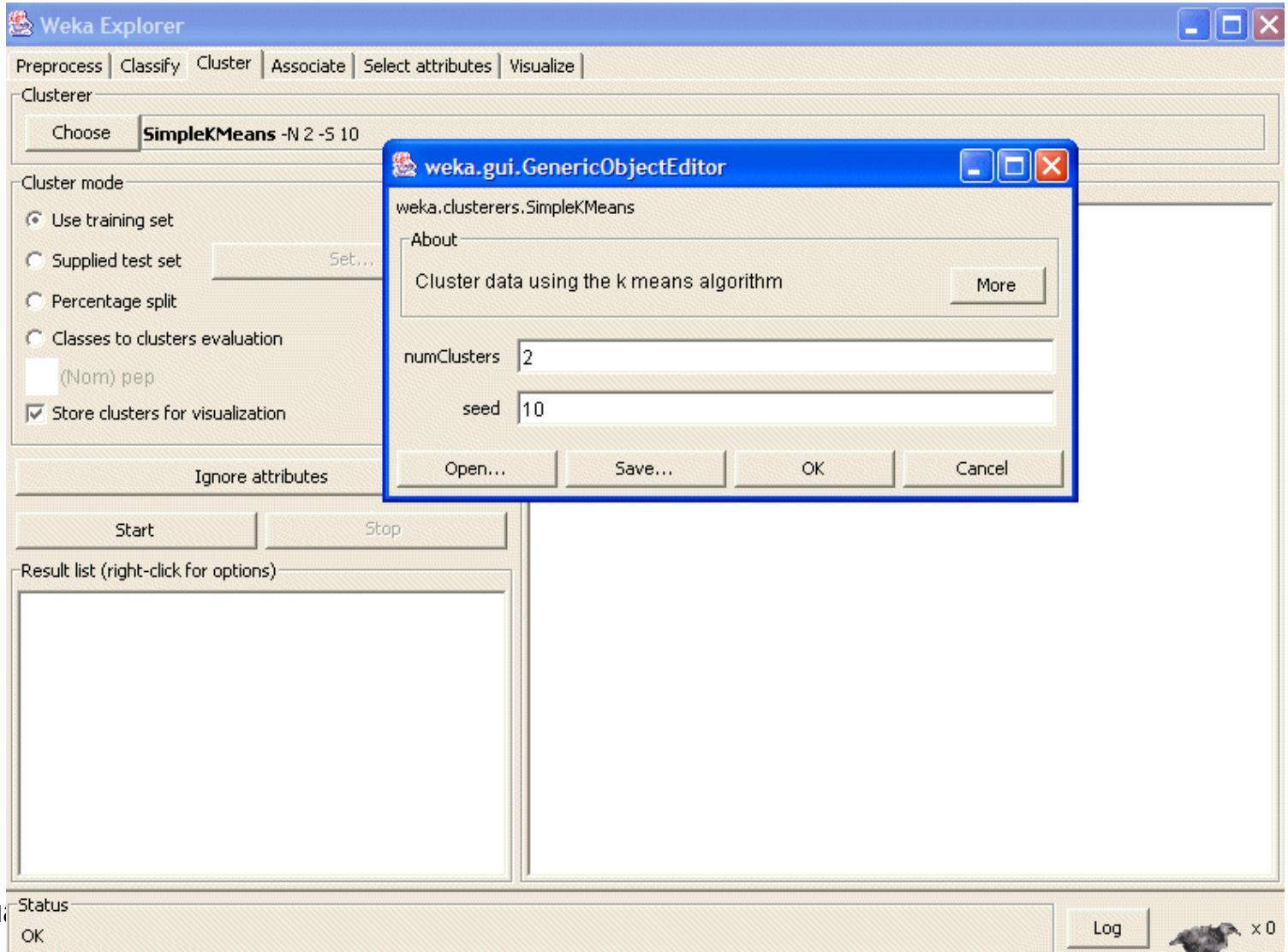
- Usiamo il file arff
  http://maya.cs.depaul.edu/classes/ect584/WEKA/cluster/bank.arff
- Contiene 600 istanze di consumatori, caratterizzati da alcune variabili e la loro preferenza rispetto al prodotto PEP.



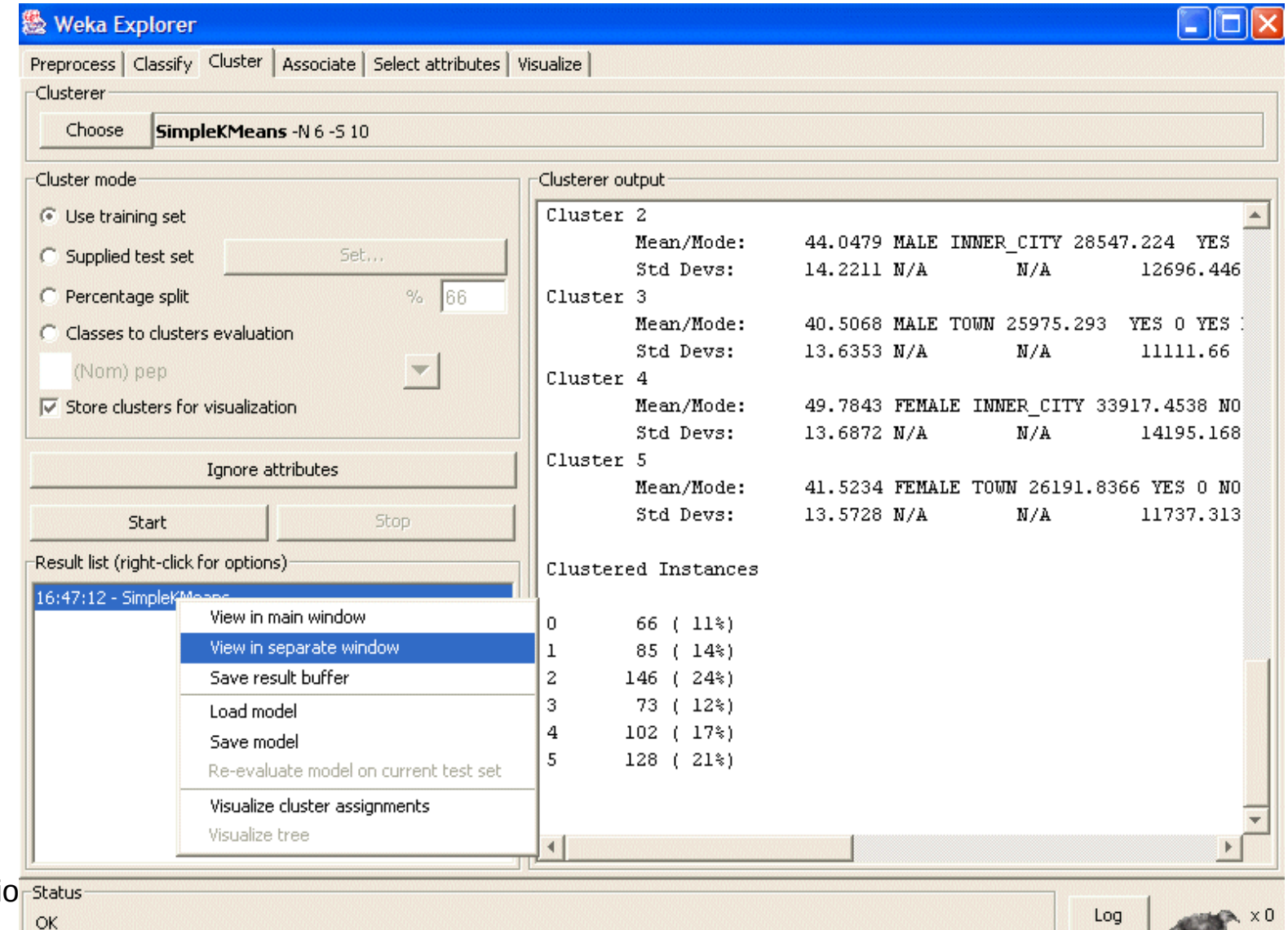
Esempio

- Alcune implementazioni di k-means consentono solo valori numerici per gli attributi.
- Può anche essere necessario normalizzare i valori degli attributi che sono misurati su scale sostanzialmente diverse (ad esempio, "età" e "reddito").
- Mentre WEKA fornisce i filtri per realizzare tutte queste attività di pre-elaborazione, non sono necessari per il clustering in WEKA.
- Questo perché SimpleKMeans gestisce automaticamente una mistura di attributi categorici e numerici.

Esempio


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 2 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
Classes to clusters evaluation
(Nom) pep
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0
weka.gui.GenericObjectEditor
weka.clusterers.SimpleKMeans
About
Cluster data using the k means algorithm
More
numClusters
2
seed
10
Open...
Save...
OK
Cancel

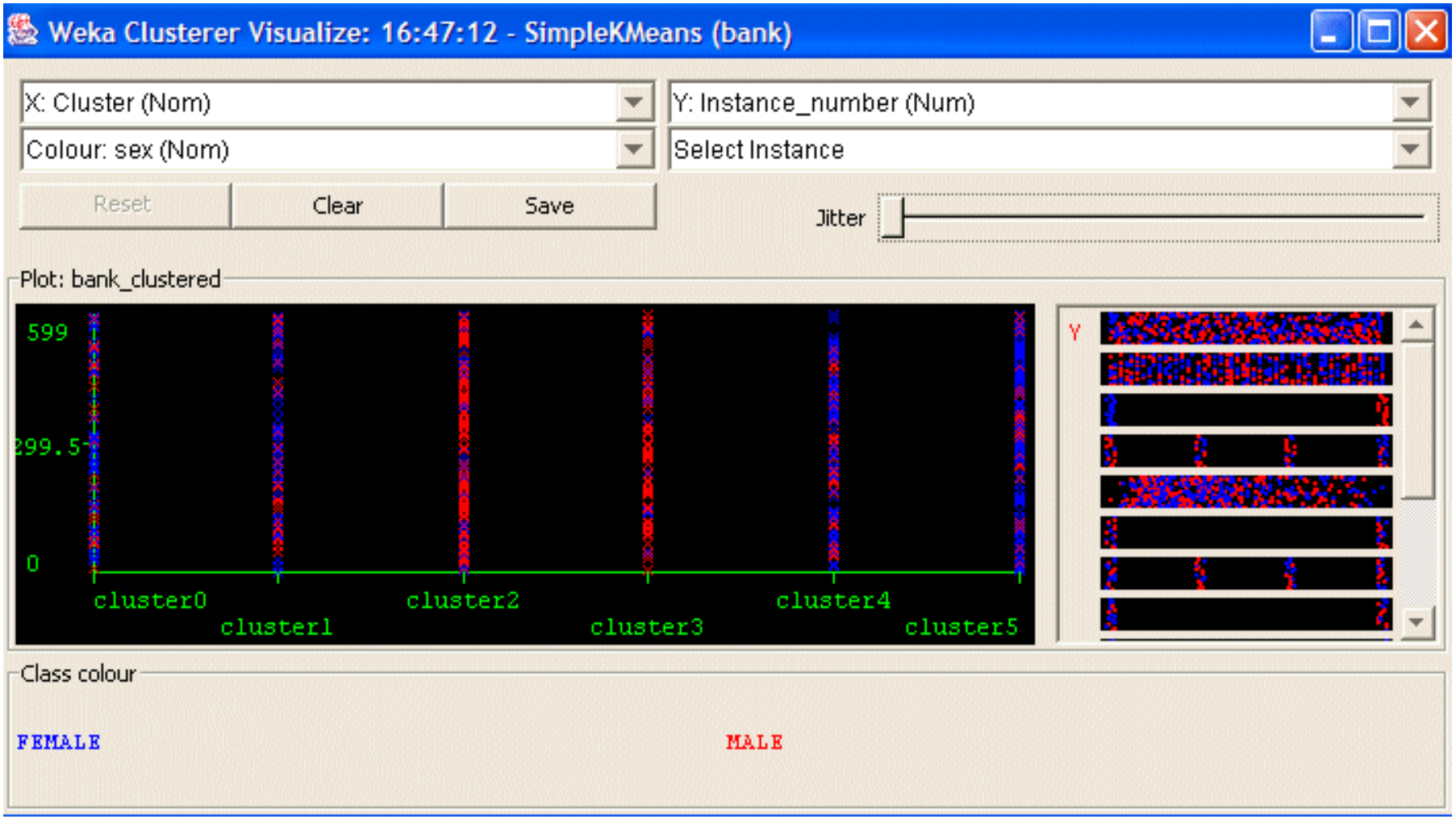
Esempio


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -N 6 -S 10
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Nom) pep
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
16:47:12 - SimpleKMeans
View in main window
View in separate window
Save result buffer
Load model
Save model
Re-evaluate model on current test set
Visualize cluster assignments
Visualize tree
Clusterer output
Cluster 2
Mean/Mode: 44.0479 MALE INNER_CITY 28547.224 YES
Std Devs: 14.2211 N/A N/A 12696.446
Cluster 3
Mean/Mode: 40.5068 MALE TOWN 25975.293 YES 0 YES
Std Devs: 13.6353 N/A N/A 11111.66
Cluster 4
Mean/Mode: 49.7843 FEMALE INNER_CITY 33917.4538 NO
Std Devs: 13.6872 N/A N/A 14195.168
Cluster 5
Mean/Mode: 41.5234 FEMALE TOWN 26191.8366 YES 0 NO
Std Devs: 13.5728 N/A N/A 11737.313
Clustered Instances
0 66 (11%)
1 85 (14%)
2 146 (24%)
3 73 (12%)
4 102 (17%)
5 128 (21%)
Status
OK
Log
x 0

Esempio

- La finestra dei risultati mostra il centroide di ogni cluster e le statistiche sul numero e percentuale di casi assegnati a gruppi diversi.
- I centroidi dei cluster sono i vettori media per ogni cluster (così, ogni valore della dimensione del baricentro rappresenta il valore medio per tale dimensione nel cluster).
- I centroidi possono essere utilizzati per caratterizzare il cluster.
- Ad esempio, il baricentro per cluster 1 mostra che questo è un segmento di casi che rappresenta per i giovani donne di mezza età (circa 38) che vivono in centro città con un reddito medio di ca. $28,500, che sono sposate con un figlio, ecc
- Inoltre, questo gruppo hanno, in media, ha detto SI 'al prodotto PEP.

Esempio


Weka Clusterer Visualize: 16:47:12 - SimpleKMeans (bank)
X: Cluster (Nom)
Y: Instance_number (Num)
Colour: sex (Nom)
Select Instance
Reset
Clear
Save
Jitter
Plot: bank_clustered
599
299.5
0
cluster0
cluster1
cluster2
cluster3
cluster4
cluster5
Y
Class colour
FEMALE
MALE

Esempio

- È possibile scegliere il numero di cluster e uno qualsiasi degli altri attributi per ognuna delle tre diverse dimensioni disponibili (asse x, asse y, e il colore).
- Diverse combinazioni di scelte si tradurranno in una resa visiva dei rapporti differenti all'interno di ogni cluster.
- Nell'esempio precedente, abbiamo scelto il numero di cluster come l'asse x, il numero di istanza, come l'asse y, e il sesso "attributo", come la dimensione del colore.
- Questo si tradurrà in una visualizzazione della distribuzione di maschi e femmine in ogni cluster.
- Per esempio, si può notare che i cluster 2 e 3 sono dominati dai maschi, mentre i raggruppamenti di 4 e 5 sono dominati dalle femmine.

FINE