File: /General/MLAB-Text/Papers/cluster/cluster

Cluster Analysis in *MLAB*

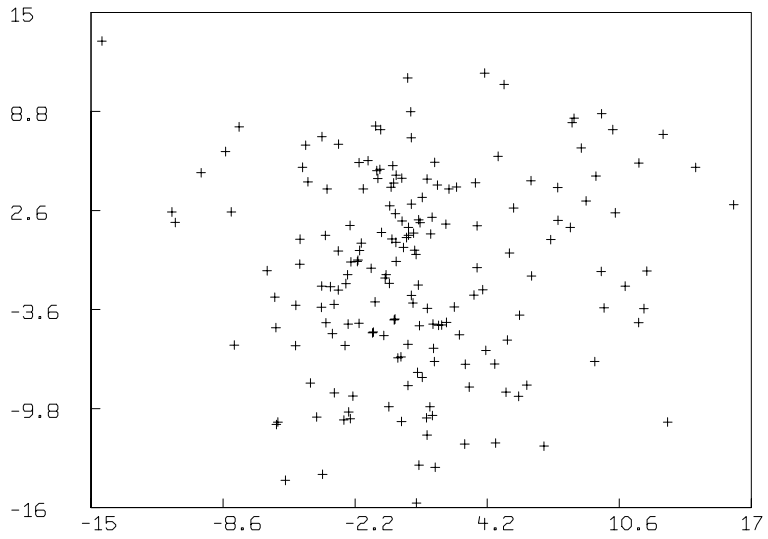
Gary D. Knott, Ph.D.
Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel. (301) 962-3711
email: csi@civilized.com URL: www.civilized.com

Situations often arise in which it is desirable to cluster a number of objects into smaller numbers of mutually exclusive groups, each having members that are as much alike as possible.

Such a clustering process depends on the 'distance' between the objects. Different definitions of 'distance' produce different clustering. *MLAB* has built-in functions for several well known clustering methods.

Below is an example of clustering using *MLAB*. The *kmeans* algorithm is used to cluster the input into 3 clusters.

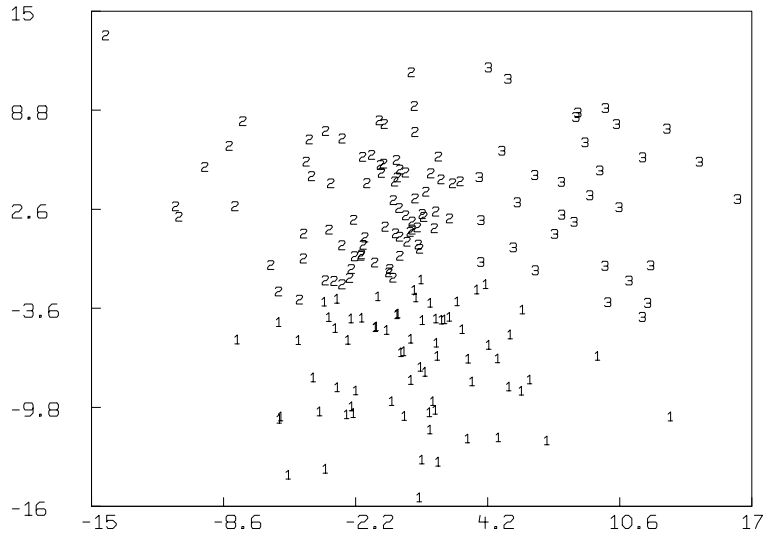
```
m = read(dataf, 500, 2);
draw m, pt crosspt lt none
view
```



```
k = kmeans(m,3)

 For 3 clusters: the initial error is 6157.127282
     after 1 iterations, the error is 4563.111736
     after 2 iterations, the error is 4364.649718
     after 3 iterations, the error is 4361.546152
     after 4 iterations, the error is 4361.546152
```

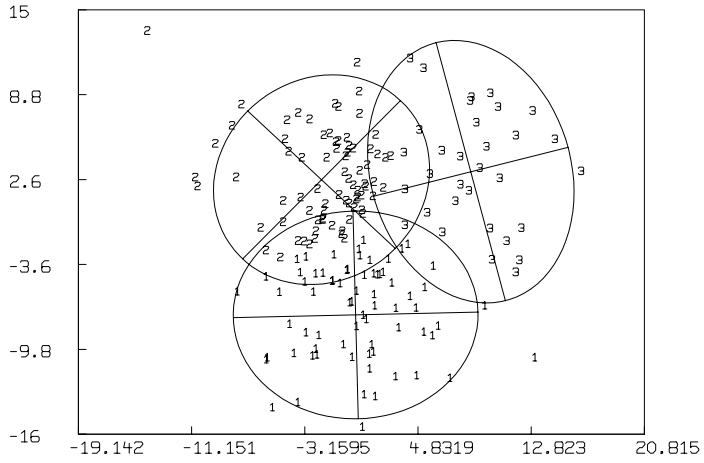
Now we will show the data points with each point labeled with its cluster number.

```
del w
draw m lt none pt none label k labelsize .01

view
```



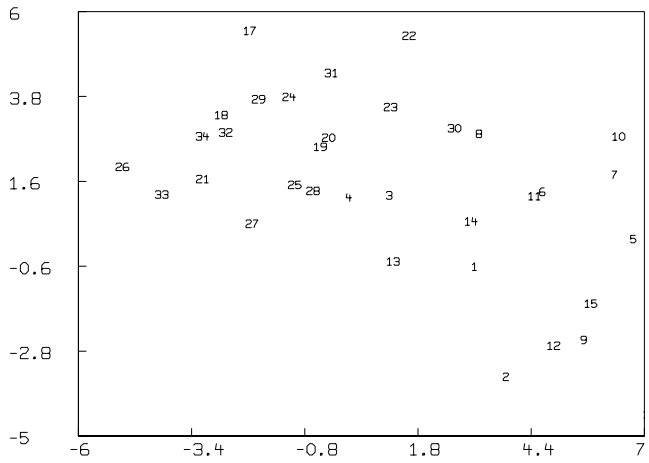
We may draw the best-fitting bivariate normal ellipical contours of each cluster as follows. We will draw the ellipses which have .68 probability content. In general the ellipse with probability content p is $\{[x_1, x_2] \mid [x_1, x_2]V^{-1}[x_1, x_2]^T = CHISQI(p, 2)\}$.

```
p = chisqi(0.68,2)
tv = 0:(2*pi)!80
fct x(t) = a*cos(t)*p
fct y(t) = b*sin(t)*p

for i = 1:3 do {
  q = compress((m&'(k=i)),3) col 1:2;
  mn = mean(q);
  c = cov(q);
  n = prcomp(c);
  a = sqrt(n[1,1]); b = sqrt(n[2,1]);

  z = (x on tv) &' (y on tv);
  e1 = n col 2:3 row 1;
  e2 = n col 2:3 row 2;
  r = e1 & e2;

  d1 = p*((-a*e1) & a*e1) + mn';
  d2 = p*((-b*e2) & b*e2) + mn';

  draw d1;
  draw d2;
  draw (z*r+mn');
}
window adjust wmatch
view
```



We can construct a so-called dendrogram in *MLAB* using various inter-cluster linking metrics. Below we will show the use of the simple centroid-based linking metric.

```
a = read(datac, 34, 2)
draw a lt none pt none label 1:34 labelsize .01
view
```




```
d = dists(a)
t = centroid(d)
n = dencurve(t)
```

```
draw n col 1:2 label n col 3:4 linetype marker labelsize .012 in w1
top title "centroid linkage" in w1
view
```