CIKM 2003

Clustering Large and High Dimensional Data

Jacob Kogan
Charles Nicholas
Marc Teboulle

http://www.cs.umbc.edu/~nicholas/clustering

Abstract

Large and often high dimensional data sets are now increasingly common and available. Clustering techniques are used to discover natural groups in data sets, and to identify abstract structures that might reside there, without having any background knowledge of the characteristics of the data. Clustering has been used in a variety of areas, including: computer vision, VLSI design, data mining, bio–informatics (e.g. gene expression analysis), and information retrieval, to name just a few. Within IR, clustering techniques have been applied in text mining and web page clustering, among others. Indeed, clustering has been a subject of interest in IR for many years. The well–known "cluster hypothesis", for example, which says that closely associated documents tend to be relevant to the same requests (van Rijsbergen 1979) suggests that document clustering should result in more effective, as well as more efficient, retrieval. The tutorial provides an overview of clustering, and an introduction to some recently developed clustering techniques. We place particular attention on document clustering, and on the applications of modern nonlinear optimization methods.

Overview of this Tutorial

- Introduction and Basic concepts
- Hierarchical algorithms, e.g. single-link
- Examples from IR
- Clustering software

- Non-hierarchical algorithms, e.g. k-means

- Partition algorithms, e.g. spherical

- Clustering as an optimization problem

Why study clustering?

- Practical applications abound, in computer science, and other disciplines

- Clustering is an important form of abstraction: instead of referring to lots of items, refer to them as a single (aggregate) entity

- Every computer scientist should know *something* about it!

Cluster \ Year	71	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	Total
Databases, NL interfaces	8	4	1	6		5	10	1	3	5	2	5	2	4	1		3	1	1	2	2						66
General !	5	2	9	2	9	5	7	10	10	6	10	6	2	5	8	6	2	2	4	3	1		4	2	5	1	126
Models	1			2	1	1		4	1	2	1	2	1	2		2	2	2	2	3	1						30
Question answering	1			1	1	1						1					1		1			1		4	4	1	17
Syntactic phrases & SDR	1					1		1		2	1	6	3	3	2	3	2	1	1	2	1	1	3	1	1	1	37
Conceptual IR, KB IR	1			4	4	1	3	3	4	3	5	7	5	1	6	3	5	3	2	3	4	1	3	2	1	1	75
Compression	1								1	2	2	1	1	1	3	1	1		1			2			1		18
Clustering		2		1	1		2		3	3	2					1	2		1	1	2	1		1		3	26
Relevance feedback		1	1	1		2			1	1		1			1	2	4	3		1	2	1	1	1	1		25
Inverted files & implementations		1				1			1			2	1	3	1			2	1				1		1	3	18
Term weighting			1	3	2	1	2	1	1	5	3	3				1		2	1	1	1	1			1	1	31
Message understanding & TDT			1	1						1						3		2			3	4	2	4	5	5	31
Filtering			1						1			1			1			1		4	1	1	1	1	2	3	18
Hypertext IR, Multiple evidence											1	3	1	1	2	1	2	2	2	1	4	3	1	5	2	2	33
Image retrieval				1				1			1			1	1				2	1	1						9
Probabilistic & Language models				1	1	1						3	1		3	4	2	2	3	2	1	3	1		3	3	34
Boolean & extended Boolean							1		2	1				1			1	1			1		1	1			10
Japanese & Chinese IR									1					1			2		3	2	3	1	1				14
DBMS & IR				1		1		1											1	1							5
Users & Search					2		3	3	2	2	4		3	2	2	3	1	3	3	1		1	2	1			38
Visualisation										1		1	1		1			1		2	1	1	2		1		12
Signature files								1		1		1	2	2		1	1										9
Distributed IR						1		2	1		2		1		1				3	1	1	3	4	2	1	1	24
Evaluation																	3	4	4	2	1	7		2	3	8	34
Topic distillation & Linkage retrieval																						1		3	3	2	9
Latent semantic indexing												1				1		1					2	1			6
Text categorisation													1					3	3	3	1	3	1	3	3	2	23
Document summarisation																			2				2	2	3	3	12
Cross lingual																				1	3	3	1	1	3	4	16

Clustering is a hot area, in both the IR and database communities, as well as many others.

(Table on previous slide from Smeaton et al., SIGIR 2002)

Similar numbers from other conferences:

CIKM 1993,1; 95,1; 98,3; 99,4; 2000,4; 01,2; 02,5

SIGMOD 1980,1; 85,1; 90,1; 91,2; 92,1; 96,1; 98,1; 99,1; 2000,2; 02,2; 03,1

Why is clustering important in Database? or Information Retrieval?

If records can be clustered together in a sensible fashion, then indexing and retrieval operations can be optimized.

In particular, in distributed systems, records can be clustered by topic, language, source, or other attributes, and then allocated to certain nodes.

Cluster Hypothesis

The “Cluster Hypothesis”, proposed by van Rijsbergen in 1971, “states that closely associated documents tend to be relevant to the same requests.”

How well the CH applies in practice has been a research question for many years.

Perhaps more important, clustering tells you about the corpus, and what structure there is in it, before the issue of query processing even comes up.

What do we want in a clustering algorithm?

- it should produce “good” clusters, whatever that means
- reasonable performance, in terms of time and space complexity
  - scales as the number of points increases
  - scales as the dimensionality of points increases
  - amenable to parallelization

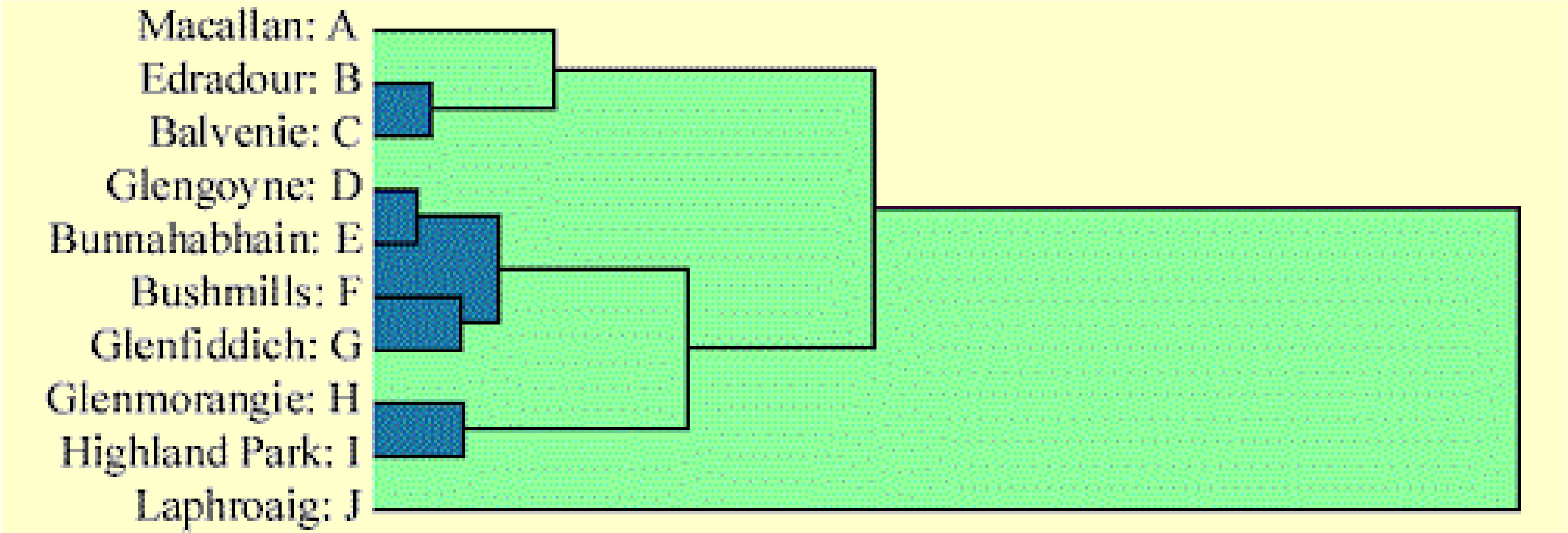
Taxonomy of clustering algorithms

(Credit to Edie Rasmussen for much of the material on these next several slides)

Hierarchical algorithms produce a nested data set

- graphical output via dendrogram, for example
- relatively slow
- “clusters within clusters”

An example dendrogram



from http://www.clustan.com

Taxonomy of clustering algorithms (continued)

Non-hierarchical algorithms produce a simple partition of the data

- graphical output via colors on a CRT, for example
- can achieve linear time and space complexity
- structure within clusters, if any, may not be apparent

Recursive application of a non-hierarchical algorithm results in a hierarchical algorithm.

Hierarchical algorithms may be *divisive*, (top-down) or *agglomerative* (bottom-up)

Agglomerative algorithms have received more attention, although a divisive hierarchical algorithm is at the heart of Scatter/Gather, for example. (Cutting et al., SIGIR 92, SIGIR 93)

Hierarchical Agglomerative Clustering Methods (HACMs)

- single link
- complete link
- group average link
- Ward’s method

Single Link

At each step, join the most similar pair of objects (clusters or single points) not yet in the same cluster. Similarity of objects is based on the similarity of their *most* similar points. Stop when the last two objects are joined, i.e. at the root of the dendrogram.

Easy to implement, and runs quickly once similarity matrix is computed.

Tends to form “long straggly” clusters.

Refer to one page of code for singleLink in handout. Also available on tutorial web site.

The code shown here runs on Octave.

```
%-------
% "singleLink"  demonstrate clustering algorithm
%-------

n = input('dimensionality of data space n (n >= 2):');
assert( n >=2 );

m = input('number of data points (vectors in R^n) to generate:');
assert( m >=2 );

% generate random points in the 0:1 hypercube
% represent the points in an m*n array called d, so that each point
% is a row of the d matrix
d = rand(m, n);

% if dimension is 2 or 3, we can do a plot

% clear the old plots, if any
```

```
if n == 2
  gplot clear
  gset noparametric
elseif n == 3
  gsplot clear
  gset parametric
else
  more off;
  page_output_immediately = 1;
endif

% plot the points in the matrix
if n <= 3
  xlabel("x axis");
  ylabel("y axis");
  zlabel("z axis");
  gset size ratio 1
  gset nokey
  gset pointsize 2
  if n == 2
    gplot [0:1] [0:1] d with points
  elseif n == 3
    gsplot [0:1] [0:1] [0:1] d with points
  endif
  % keep the same plot going through each pass
  hold on;
endif;

% now build a distance matrix, using Euclidean distance
% the matrix will be upper triangular
if n>3
  printf("building similarity matrix\n");
endif
s=zeros(m);
for i = 1:m
  for j = i:m            % note MATLAB vector op
    s(i,j) = sqrt(sumsq((d(i,:) - d(j,:)))) ;
  endfor
endfor

% loop through the data
% at each step, join the most similar pair of objects (points or clusters)
% that are not yet in the same cluster

% make a vector which says which cluster a given point is in
% initialze with each point in its own cluster

cluster = zeros(1,m);
for i=1:m
  cluster(i) = i;
```

```
endfor

if n>3
  printf("initial clustering\n");
endif

for pass = 1:m-1

% loop through the points, pairwise, and find closesti and closestj, the points
% in different clusters with the smallest distance
closestDist = n;  % bigger than is possible in n dimensional hypercube

for i=1:m;
  for j=i+1:m;
    if (cluster(i) != cluster(j)) & (s(i,j) < closestDist)
      closesti = i;
      closestj = j;
      closestDist = s(i,j);
    endif
  endfor
endfor

assert(cluster(closesti) != cluster(closestj));
%printf ("the two closest points are %d in cluster %d and %d in cluster %d\n",
% closestj,cluster(closestj), closesti, cluster(closesti));
```

```
newLink = zeros(2,n);
newLink(1,:) = d(closesti,:);
newLink(2,:) = d(closestj,:);

oldc = cluster(closestj);
newc = cluster(closesti);

if n == 2
  gplot newLink with lines
elseif n == 3
  gsplot newLink with lines
else
  printf("merge cluster %d into cluster %d\n", oldc, newc);
endif

% connect the clusters by making each point in closesti's cluster a member of
% closestj's cluster

for k=1:m
  if cluster(k) == oldc
    cluster(k) = newc;
  endif
endfor

endfor
```

Complete Link

At each step, join the most similar pair of objects not yet in the same cluster. Similarity of objects is based on the similarity of their *least* similar points. Stop when the last two objects are joined, i.e. at the root of the dendrogram.

Implementation is slightly more complicated than single link, but performance is about the same.

Tends to form compact clusters.

Code example available on tutorial web site.

The Similarity Matrix

Hierarchical algorithms often need to know the similarity between each pair of points in the data.

In document clustering, using cosine similarity, documents with no terms in common have similarity zero. Therefore, the density of the similarity matrix depends on the data. Furthermore, the similarity matrix will need to be modified as clusters are created in each pass of the algorithm.

To build the similarity matrix obviously requires $O(n^2)$ time and space in a naive implementation. Some algorithms allow for similarity computations on the fly, and therefore require a little more time, but significantly less space.

Comparing single link vs. complete link

Code is available on tutorial web site. (Demo Octave, if time and facilities permit)

```
start octave
singleLink
completeLink
compareLink
```

On random data, the two algorithms seem to produce very similar results.

Group Average Link

Like single link and complete link, except that similarity of objects is based on average similarity of component points.

Implementation very much like complete link.

Ward’s Method

Like the single link, complete link, and group average link, except that at each step, the two objects being joined result in the smallest increase in total intra-cluster sum of squares.

“tends to produce homogeneous clusters and a symmetric hierarchy”

Evaluation of HACMs

The algorithms are fairly similar, differing mostly in how they calculate the similarity between two clusters.

All rely on calculating (perhaps on-the-fly), and perhaps recalculating, a similarity matrix.

Complexity $O(n^2 logn)$

The Lance-Williams dissimilarity update formula lets an algorithm calculate the similarities between newly formed clusters.

Improvements have been made to various HACMs, especially single link, for which several optimal algorithms (i.e. $O(n^2)$ time and $O(n)$ space) are known. These include SLINK and Prim-Dijkstra.

Algorithms for Non-hierarchical methods

Single pass is simple, but sensitive to order of input data, e.g. batch k-means

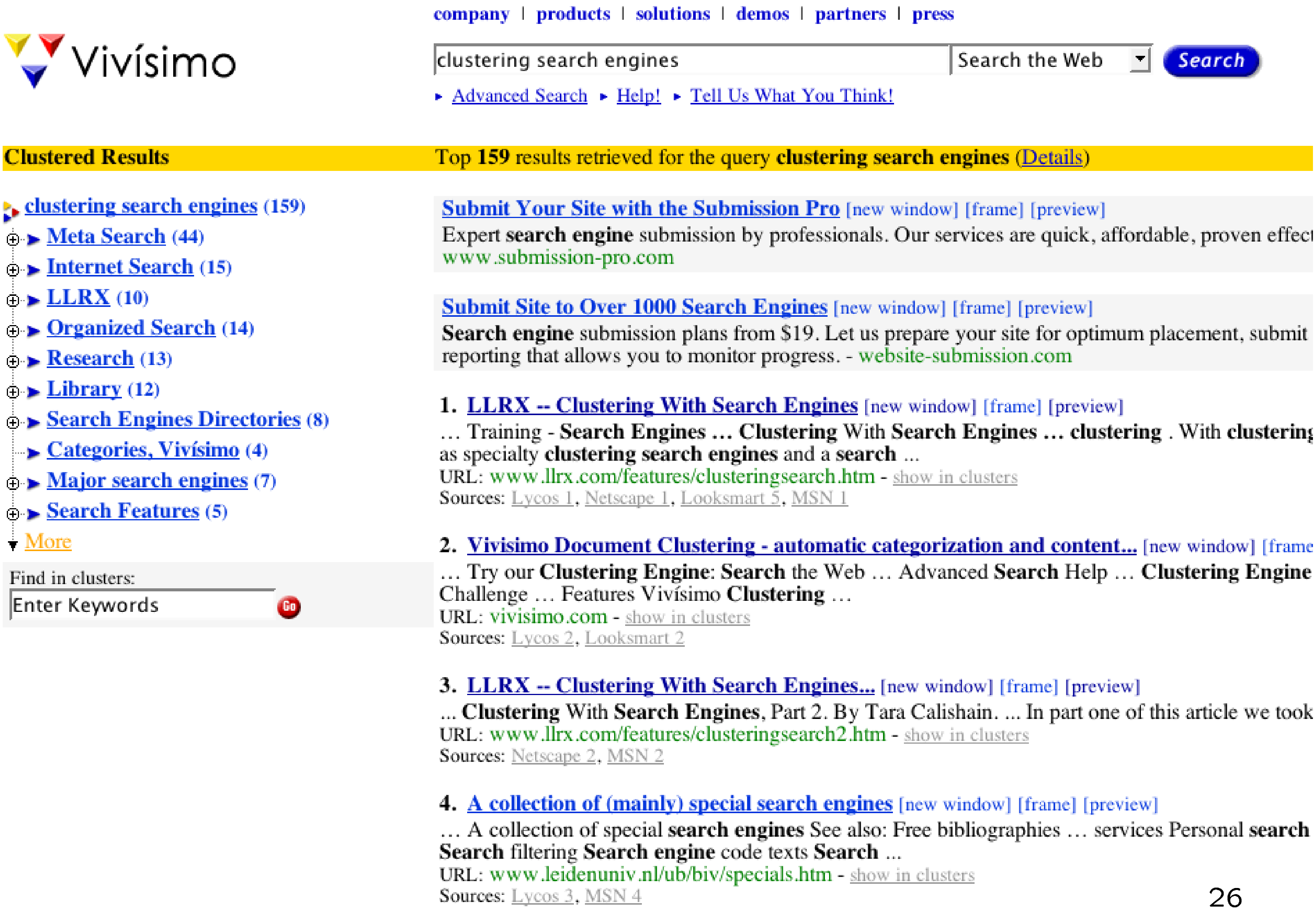
Reallocation lets documents move from one cluster to another, e.g incremental k-means

Uses of clustering in IR

- **Query Processing**: Given a query q, work down the tree, taking branches where similarity with the query is greatest. Continue until some stopping criterion is met, e.g. subcollection is a single document, or if the similarity starts to decrease.

- **Results Presentation**: Process query q as usual, but cluster the results. Some search engines do this, but not very many. See (or demo) http://www.vivisimo.com

- **Exploring the Corpus**: Find clusters in order to better understand what documents (or topics or languages) are available.


Vivísimo
company | products | solutions | demos | partners | press
clustering search engines
Search the Web
Search
▸ Advanced Search ▸ Help! ▸ Tell Us What You Think!

Clustered Results

- **clustering search engines (159)**
  - ⊕ ▶ Meta Search (44)
  - ⊕ ▶ Internet Search (15)
  - ⊕ ▶ LLRX (10)
  - ⊕ ▶ Organized Search (14)
  - ⊕ ▶ Research (13)
  - ⊕ ▶ Library (12)
  - ⊕ ▶ Search Engines Directories (8)
  - ▶ Categories, Vivísimo (4)
  - ⊕ ▶ Major search engines (7)
  - ⊕ ▶ Search Features (5)
  - More

Find in clusters:
Enter Keywords
Go

Top **159** results retrieved for the query **clustering search engines** (Details)

Submit Your Site with the Submission Pro [new window] [frame] [preview]
Expert **search engine** submission by professionals. Our services are quick, affordable, proven effect
www.submission-pro.com

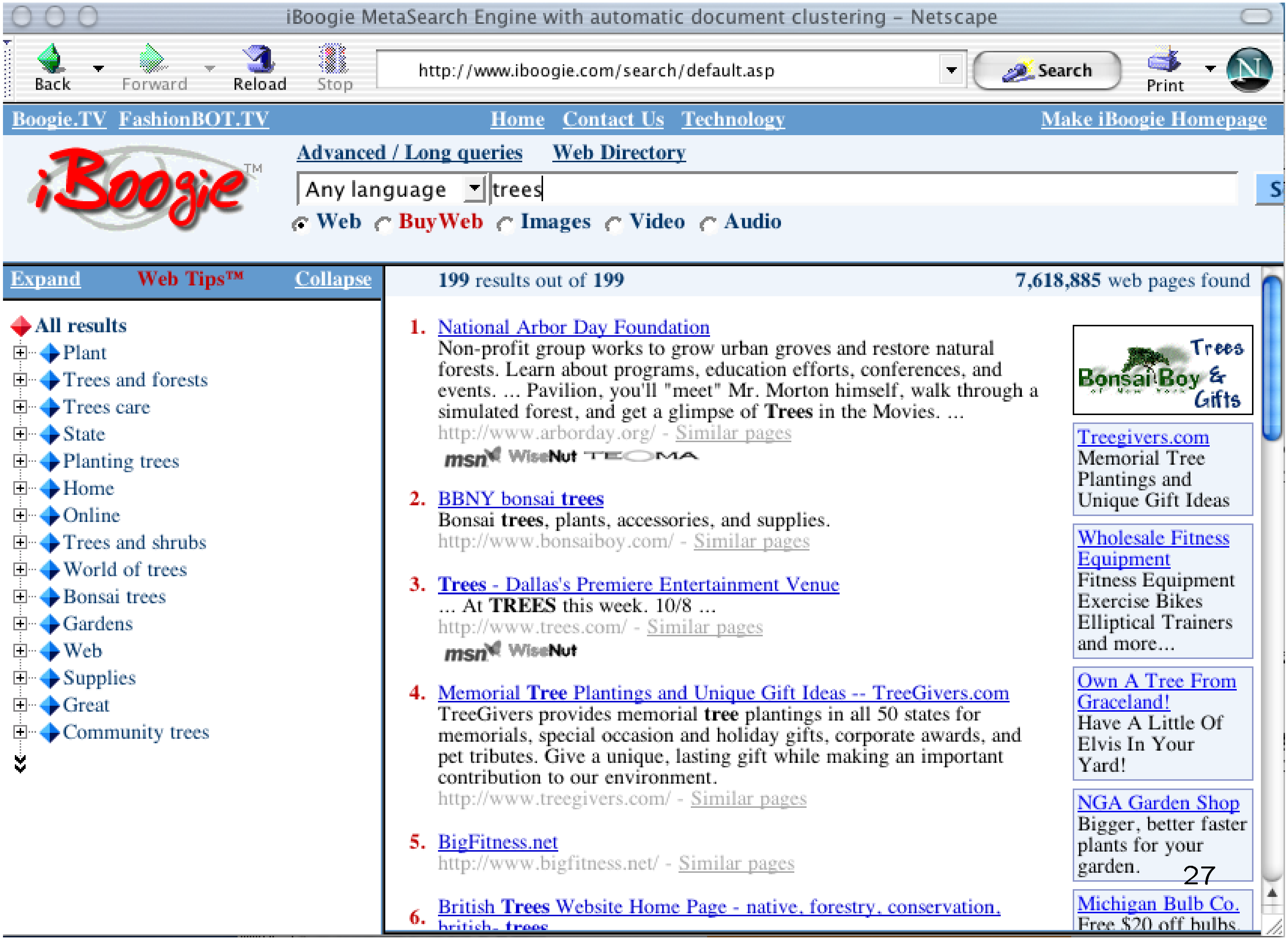
Submit Site to Over 1000 Search Engines [new window] [frame] [preview]
Search engine submission plans from $19. Let us prepare your site for optimum placement, submit reporting that allows you to monitor progress. - website-submission.com

1. **LLRX -- Clustering With Search Engines** [new window] [frame] [preview]
… Training - **Search Engines … Clustering** With **Search Engines … clustering** . With **clustering** as specialty **clustering search engines** and a **search** ...
URL: www.llrx.com/features/clusteringsearch.htm - show in clusters
Sources: Lycos 1, Netscape 1, Looksmart 5, MSN 1

2. **Vivisimo Document Clustering - automatic categorization and content...** [new window] [frame
… Try our **Clustering Engine**: **Search** the Web … Advanced **Search** Help … **Clustering Engine** Challenge … Features Vivísimo **Clustering** …
URL: vivisimo.com - show in clusters
Sources: Lycos 2, Looksmart 2

3. **LLRX -- Clustering With Search Engines...** [new window] [frame] [preview]
... **Clustering** With **Search Engines**, Part 2. By Tara Calishain. ... In part one of this article we took
URL: www.llrx.com/features/clusteringsearch2.htm - show in clusters
Sources: Netscape 2, MSN 2

4. **A collection of (mainly) special search engines** [new window] [frame] [preview]
… A collection of special **search engines** See also: Free bibliographies … services Personal **search** **Search** filtering **Search engine** code texts **Search** ...
URL: www.leidenuniv.nl/ub/biv/specials.htm - show in clusters
Sources: Lycos 3, MSN 4



iBoogie MetaSearch Engine with automatic document clustering – Netscape

http://www.iboogie.com/search/default.asp

Boogie.TV FashionBOT.TV Home Contact Us Technology Make iBoogie Homepage

Advanced / Long queries Web Directory

Any language | trees

● Web ○ BuyWeb ○ Images ○ Video ○ Audio

Expand Web Tips™ Collapse

- All results
  - Plant
  - Trees and forests
  - Trees care
  - State
  - Planting trees
  - Home
  - Online
  - Trees and shrubs
  - World of trees
  - Bonsai trees
  - Gardens
  - Web
  - Supplies
  - Great
  - Community trees

199 results out of 199 — 7,618,885 web pages found

1. National Arbor Day Foundation
Non-profit group works to grow urban groves and restore natural forests. Learn about programs, education efforts, conferences, and events. ... Pavilion, you'll "meet" Mr. Morton himself, walk through a simulated forest, and get a glimpse of **Trees** in the Movies. ...
http://www.arborday.org/ - Similar pages
msn WiseNut TEOMA

2. BBNY bonsai **trees**
Bonsai **trees**, plants, accessories, and supplies.
http://www.bonsaiboy.com/ - Similar pages

3. **Trees** - Dallas's Premiere Entertainment Venue
... At **TREES** this week. 10/8 ...
http://www.trees.com/ - Similar pages
msn WiseNut

4. Memorial **Tree** Plantings and Unique Gift Ideas -- TreeGivers.com
TreeGivers provides memorial **tree** plantings in all 50 states for memorials, special occasion and holiday gifts, corporate awards, and pet tributes. Give a unique, lasting gift while making an important contribution to our environment.
http://www.treegivers.com/ - Similar pages

5. BigFitness.net
http://www.bigfitness.net/ - Similar pages

6. British **Trees** Website Home Page - native, forestry, conservation, british- trees

Smeaton's SIGIR paper study

- Abstracts from 853 SIGIR papers were collected.
- Stopwords were removed, and terms stemmed and weighted, and similarity matrix calculated
- Document vectors were clustered using a hierarchical, agglomerative method.
- Similarity threshold was adjusted to produce a desirable number of clusters, which were then labeled by inspection

What about the whisky example?

from http://www.whiskyclassified.com/

- Twelve characteristics of whisky were identified

- Reviews of 86 different single malt whiskies were collected, and the reviewers' ratings were noted

- The clustering software grouped whiskies into the same cluster when they have broadly the same taste characteristics across all 12 sensory variables... [the algorithm] minimizes the variance within clusters and maximizes the variance between clusters.

Scatter/Gather

Key idea: clustering can be effective as an access method in its own right, apart from retrieval.

- Data is divided into a small number of clusters
- Short summaries are presented to user, who selects one or more clusters for further study
- Those clusters are gathered, clustered, and the process repeats.

The process is user-driven, divisive (i.e. top-down) and hierarchical. Since system is interactive, speed of clustering is critical.

Buckshot Algorithm

- Take a random sample of the data of size $\sqrt(kn)$, where k is the number of clusters, and n is the total number of documents

- Find k "centers" in the sample using e.g. group average link

- Assign each document to one of the clusters, e.g. by closest distance

- Cluster centers may shift, so repeat assignment once or twice.

Fractionation Algorithm

- Divide the document set into N/m groups of fixed size m, with $m > k$.

- Cluster the data in each of the buckets using some clustering algorithm. Treat these N/m groups as individuals, and repeat, until only k groups remain

- Assign each document to one of these k clusters, as in buckshot

- Cluster centers may shift, so repeat assignment once or twice

Example: start with $N = 100000$ documents. To choose $k = 10$ centers, let $m = 100$.

Make $N/m = 1000$ document sets, 100 documents in each set. Find the “center” each of these sets, using e.g. group average link. Cost: $O(100^2)$, with a factor of 1000.

Treating these 1000 “centers” as individuals, divide them into 10 sets of 100. Find the centers of these 10 sets, and the resulting 10 cluster centers are what we wanted. Cost: $O(100^2)$, with a factor of 10.

Total cost is still $O(n^2)$ for $n = 100$, which is a million times better than $O(n^2)$ for $n = 100000$

Fractionation vs. Buckshot

Fractionation takes longer than buckshot, although asymptotic complexity is the same, but doesn't have the randomness of buckshot.

Buckshot can be run several times, in the hopes of getting better results.

According to Cutting et al, Fractionation seems to make better clusters

Stratified vs. random sampling is perhaps even more important for documents than for widgets :-)

Star Clusters - another IR application of clustering

A hierarchical technique for browsing an information collection.

Each level in the hierarchy is determined by a minimum similarity threshold between pairs of documents.

Find star-shaped subgraphs such that each document is one (and possibly more than one!) subgraph, where the distance between each “satellite” document and the “star” at the cluster’s center is greater than the minimum for that level of the hierarchy.

From Jain’s 1999 Survey on Data Clustering

Components of a Clustering Task:

1. Choose a representation of the data
2. Define a similarity measure
3. Clustering algorithm
4. Data abstraction, if needed
5. Assess output, if needed (and it usually is - CKN)

Data representation

- Feature selection
- Dimension reduction
- Transformation e.g. Cartesian to polar coordinates for points in R^2

Speaking of Dimension Reduction...

High-dimensionality can be an issue

Documents “live” in high dimensional spaces, and they may have similarities that the vector space representation misses, e.g. no terms in common, but term frequencies are identical. How could this happen?

Lots of recent work on projecting data into lower dimensional spaces, then cluster.

Similarity measures

For document clustering, one can use Cosine, Dice, or Jaccard similarity measures

$$S_{D_i,D_j} = \frac{\sum_{k=1}^{L} weight_{i,k} weight_{j,k}}{\sqrt{\sum_{k=1}^{L} weight_{i,k}^2 \sum_{k=1}^{L} weight_{j,k}^2}} \tag{1}$$

For semi-structured data, e.g. text with nominal or ordinal valued metadata, similarity measures that account for the different types of data have been developed. (Jain, pg. 272)

Similarity vs. distance

Euclidean distance is another reasonable choice in many applications, including (length-normalized) documents when represented as points on a hyper-sphere.

A metric space is a space in which a distance metric has the properties of
strict positiveness, $d(x, y) = 0 iff x = y, \forall x, y,$
symmetry, $d(a, b) = d(b, a) \forall a, b$
and the triangle inequality $d(a, b) + d(b, c) \geq d(a, c) \forall a, b, c$

However, measures such as Mutual Neighbor Distance do not satisfy the Triangle Inequality, but may still produce good results.

Assessing the Output: what is a “good” clustering?

How does one measure the quality of a clustering?

From an objective standpoint, cluster quality is usually expressed as an optimization, e.g. find the cluster centers that maximize total intra-cluster similarity while minimizing total inter-cluster similarity.

Image processing papers use subjective assessment of test images, as well as objective functions.

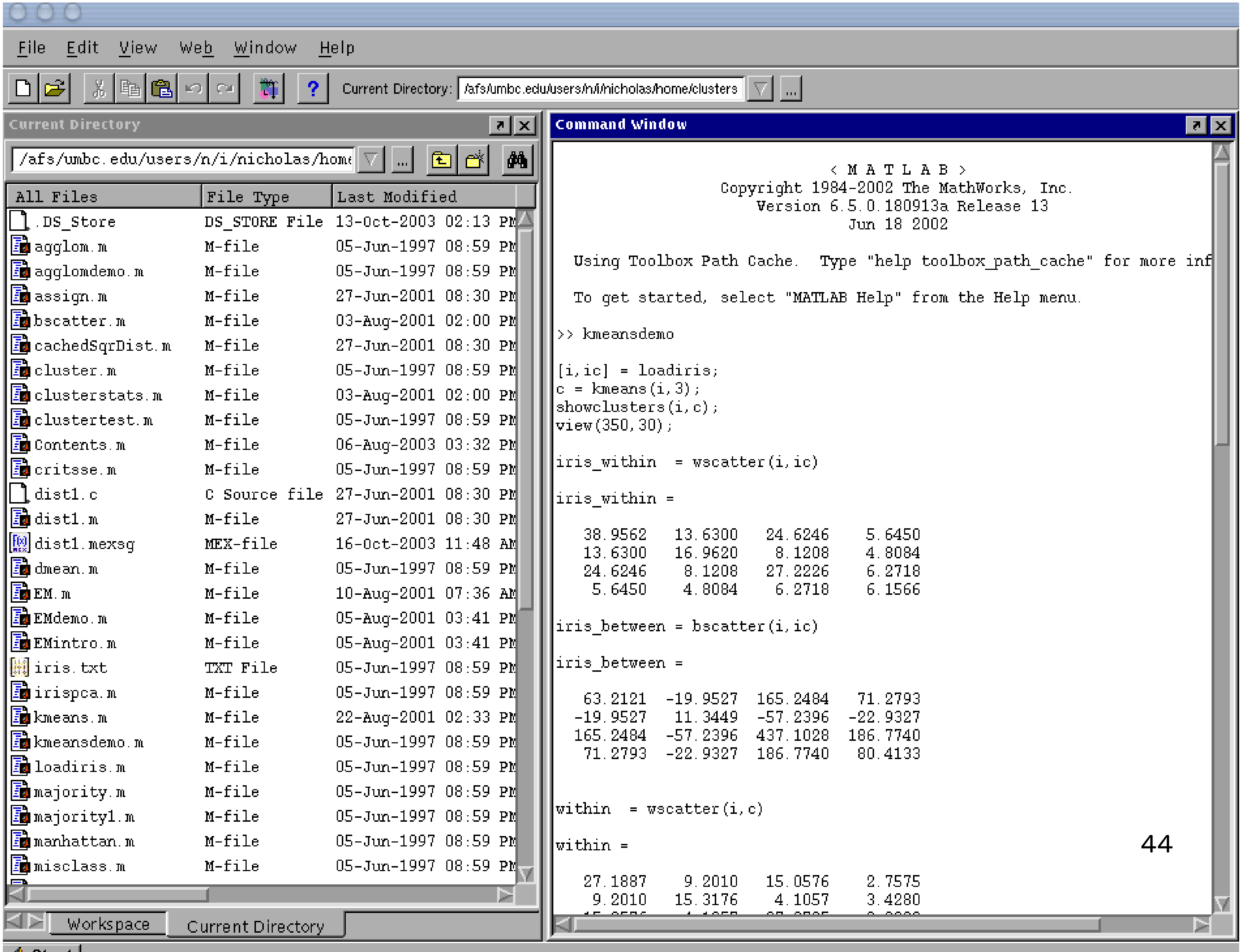
Using clusters package in Matlab

Get the package clusters from Dellaert’s site.

Start Matlab, and then type “kmeansdemo” in command window.

Matlab files in this Directory

projectpca	PROJECTPCA : project data matrix on first nr eigenvectors
showpca2	SHOWPCA2 : project data matrix on 2 first eigenvectors and show them
critsse	CRITSSE : computes Sum-of-Squared-Error Criterion for a given clustering
showpca3	SHOWPCA3 : project data matrix on 3 first eigenvectors and show them
wscatter	WSCATTER(x,c) = within-cluster scatter matrix
clustertest	CLUSTERTEST : test clusterstats with really simple distribution
agglomdemo	AGGLOMDEMO : demonstrate agglomerative clustering
clusterstats	CLUSTERSTATS(x,c) computes the statistics for each cluster
cluster	CLUSTER : return the matrix of samples in cluster j according to c
bscatter	BSCATTER : between-cluster scatter matrix
agglom	AGGLOM : Basic Agglomerative Clustering
misclass	MISCLASS : calculates percent of misclassified samples in clusters
misclass1	MISCLASS1 : calculates % misclassified samples in a cluster with respect to maj. vote
loadiris	LOADIRIS : loads the cluster IRIS benchmark data
kmeansdemo	KMEANSDEMO: demonstrate k-means clustering
kmeans	KMEANS : k-means clustering
majority	MAJORITY : returns (weighted) majority vote
majority1	MAJORITY1: returns weighted majority vote for a *row vector*
printClusters	PRINTCLUSTERS : print out j-component of the data in each cluster
irispca	irispca: show first two principal components of iris data
dmean	DMEAN : distance between means of two clusters
sqrDist	SQRDIST : calculate a 1*n vector D containing the squared distances from z
showclusters	SHOWCLUSTERS: project data matrix on first eigenvectors (if necessary)
scatter	SCATTER : scatter matrix for samples x
nearest	NEAREST: return the vector zj in z that is nearest to xi
move	MOVE: move sample x(s) from its current cluster c(s) to cluster j



File Edit View Web Window Help
Current Directory: /afs/umbc.edu/users/n/i/nicholas/home/clusters
Current Directory
/afs/umbc.edu/users/n/i/nicholas/home
Command Window
Workspace
Current Directory

All Files	File Type	Last Modified
.DS_Store	DS_STORE File	13-Oct-2003 02:13 PM
agglom.m	M-file	05-Jun-1997 08:59 PM
agglomdemo.m	M-file	05-Jun-1997 08:59 PM
assign.m	M-file	27-Jun-2001 08:30 PM
bscatter.m	M-file	03-Aug-2001 02:00 PM
cachedSqrDist.m	M-file	27-Jun-2001 08:30 PM
cluster.m	M-file	05-Jun-1997 08:59 PM
clusterstats.m	M-file	03-Aug-2001 02:00 PM
clustertest.m	M-file	05-Jun-1997 08:59 PM
Contents.m	M-file	06-Aug-2003 03:32 PM
critsse.m	M-file	05-Jun-1997 08:59 PM
dist1.c	C Source file	27-Jun-2001 08:30 PM
dist1.m	M-file	27-Jun-2001 08:30 PM
dist1.mexsg	MEX-file	16-Oct-2003 11:48 AM
dmean.m	M-file	05-Jun-1997 08:59 PM
EM.m	M-file	10-Aug-2001 07:36 AM
EMdemo.m	M-file	05-Aug-2001 03:41 PM
EMintro.m	M-file	05-Aug-2001 03:41 PM
iris.txt	TXT File	05-Jun-1997 08:59 PM
irispca.m	M-file	05-Jun-1997 08:59 PM
kmeans.m	M-file	22-Aug-2001 02:33 PM
kmeansdemo.m	M-file	05-Jun-1997 08:59 PM
loadiris.m	M-file	05-Jun-1997 08:59 PM
majority.m	M-file	05-Jun-1997 08:59 PM
majority1.m	M-file	05-Jun-1997 08:59 PM
manhattan.m	M-file	05-Jun-1997 08:59 PM
misclass.m	M-file	05-Jun-1997 08:59 PM

```
                           < M A T L A B >
               Copyright 1984-2002 The MathWorks, Inc.
                   Version 6.5.0.180913a Release 13
                             Jun 18 2002

  Using Toolbox Path Cache.  Type "help toolbox_path_cache" for more inf

  To get started, select "MATLAB Help" from the Help menu.

>> kmeansdemo

[i,ic] = loadiris;
c = kmeans(i,3);
showclusters(i,c);
view(350,30);

iris_within  = wscatter(i,ic)

iris_within =

   38.9562   13.6300   24.6246    5.6450
   13.6300   16.9620    8.1208    4.8084
   24.6246    8.1208   27.2226    6.2718
    5.6450    4.8084    6.2718    6.1566

iris_between = bscatter(i,ic)

iris_between =

   63.2121  -19.9527  165.2484   71.2793
  -19.9527   11.3449  -57.2396  -22.9327
  165.2484  -57.2396  437.1028  186.7740
   71.2793  -22.9327  186.7740   80.4133


within  = wscatter(i,c)

within =

   27.1887    9.2010   15.0576    2.7575
    9.2010   15.3176    4.1057    3.4280
```

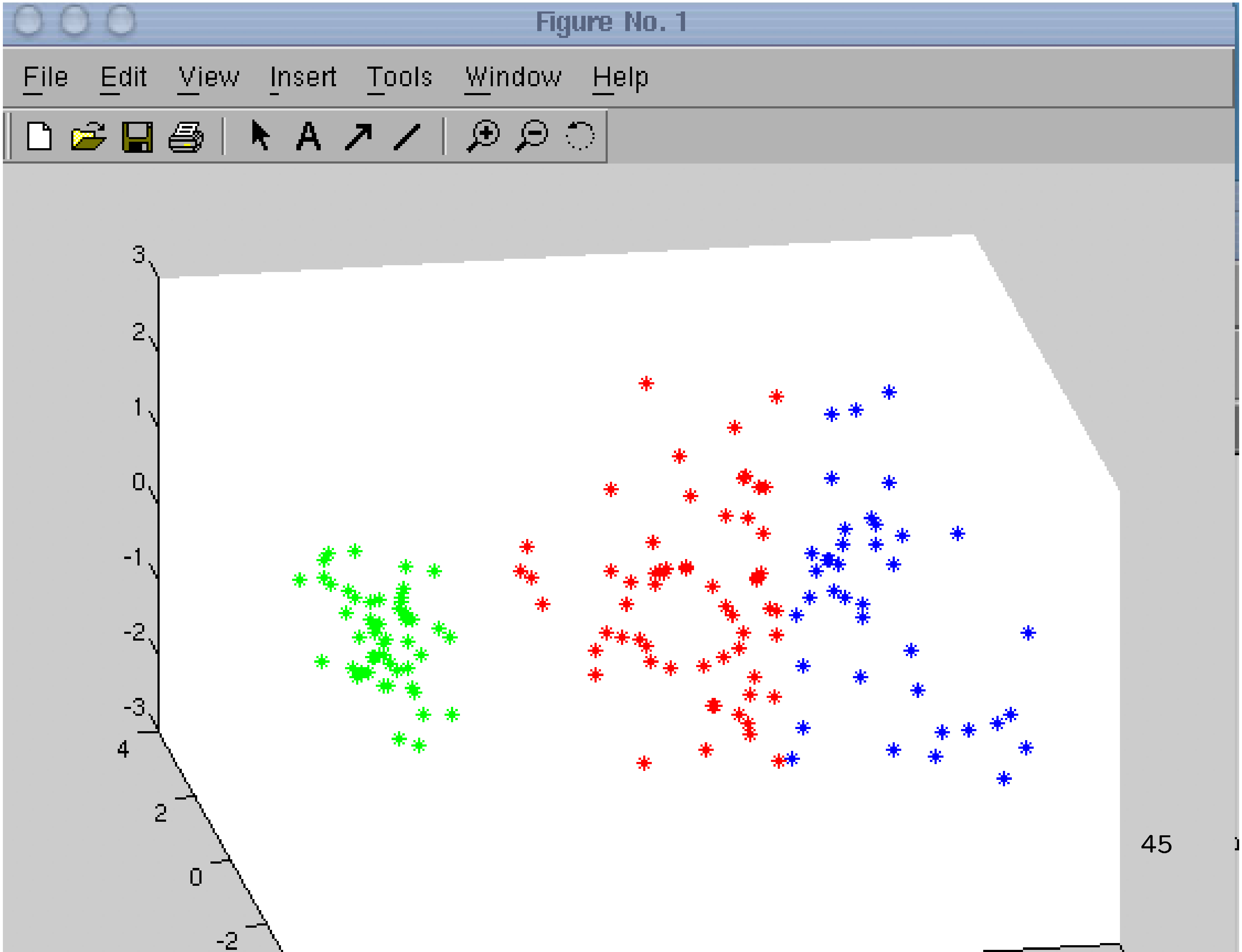
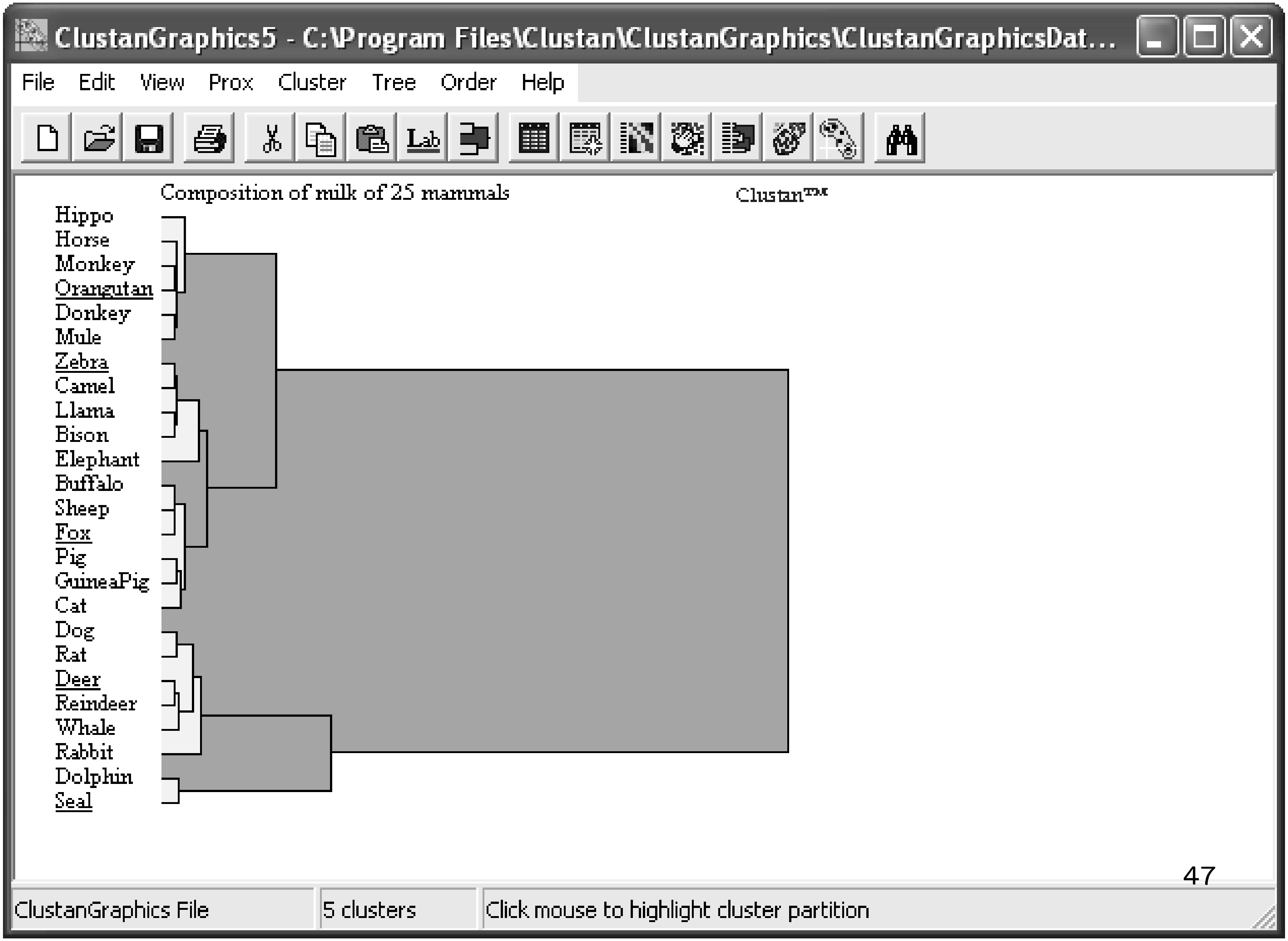


Figure No. 1
File
Edit
View
Insert
Tools
Window
Help
3
2
1
0
-1
-2
-3
4
2
0
-2
45

Using Clustan

Need a Windows platform. Click on Clustan Graphics 6 icon. Under the file menu, open (or reopen) mammals. We can demonstrate varying number of clusters, and pivoting of data.

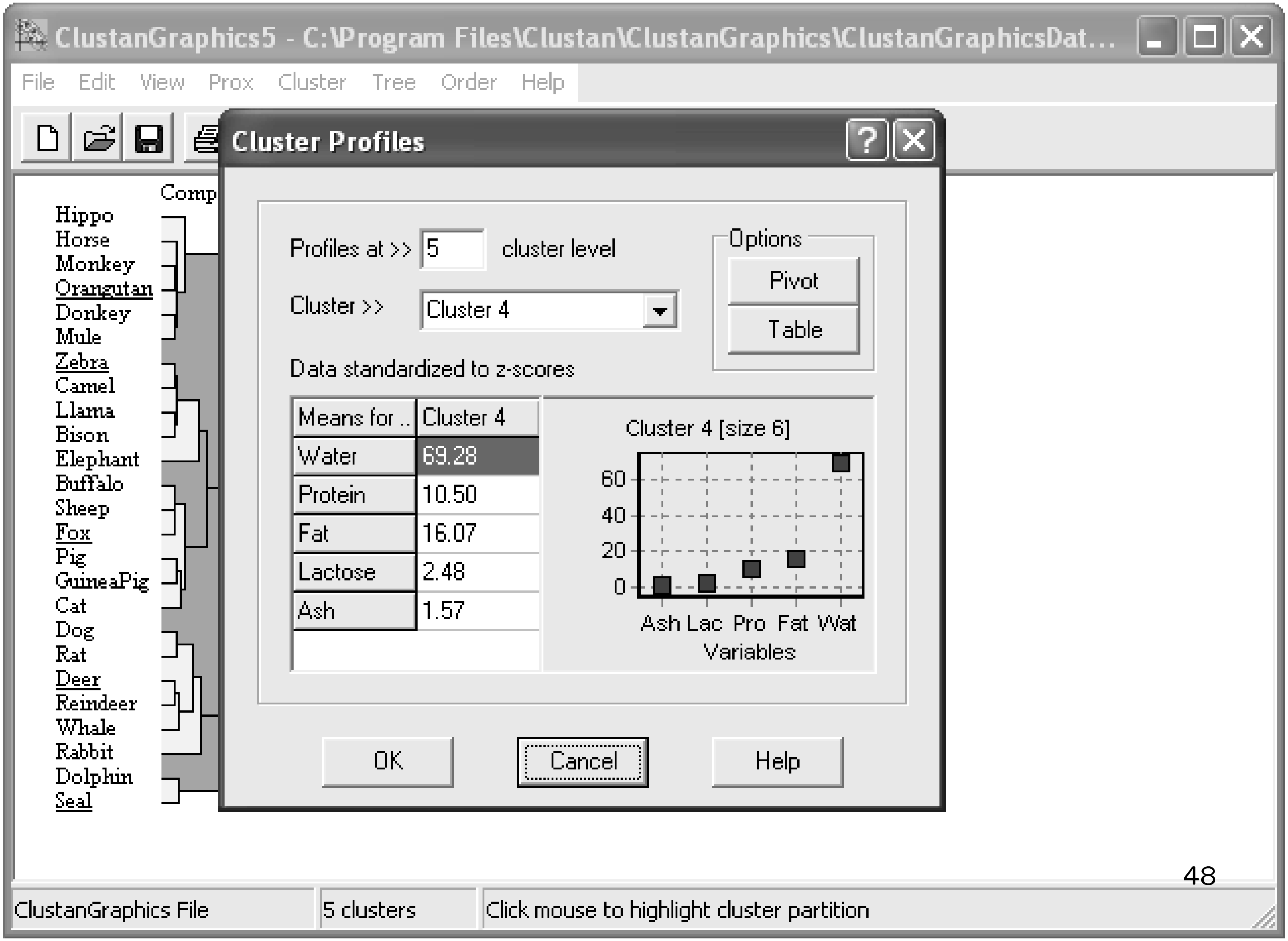
Pivoting refers to looking at variables across clusters, or looking at clusters across variables.



ClustanGraphics File

5 clusters

Click mouse to highlight cluster partition



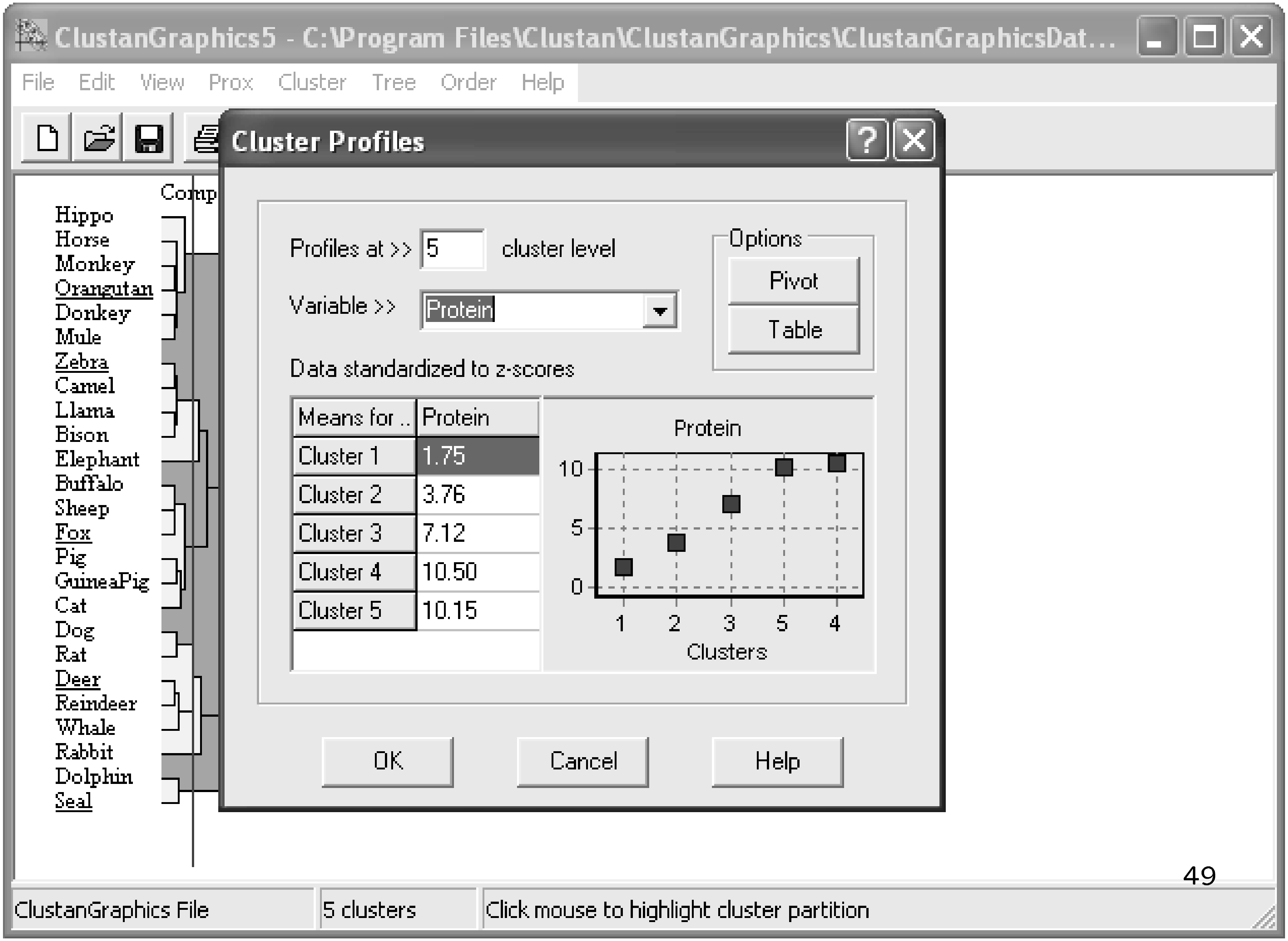
Profiles at >> 5 cluster level

Cluster >> Cluster 4

Data standardized to z-scores

Means for ..	Cluster 4
Water	69.28
Protein	10.50
Fat	16.07
Lactose	2.48
Ash	1.57

ClustanGraphics File | 5 clusters | Click mouse to highlight cluster partition



Profiles at >> 5 cluster level

Variable >> Protein

Data standardized to z-scores

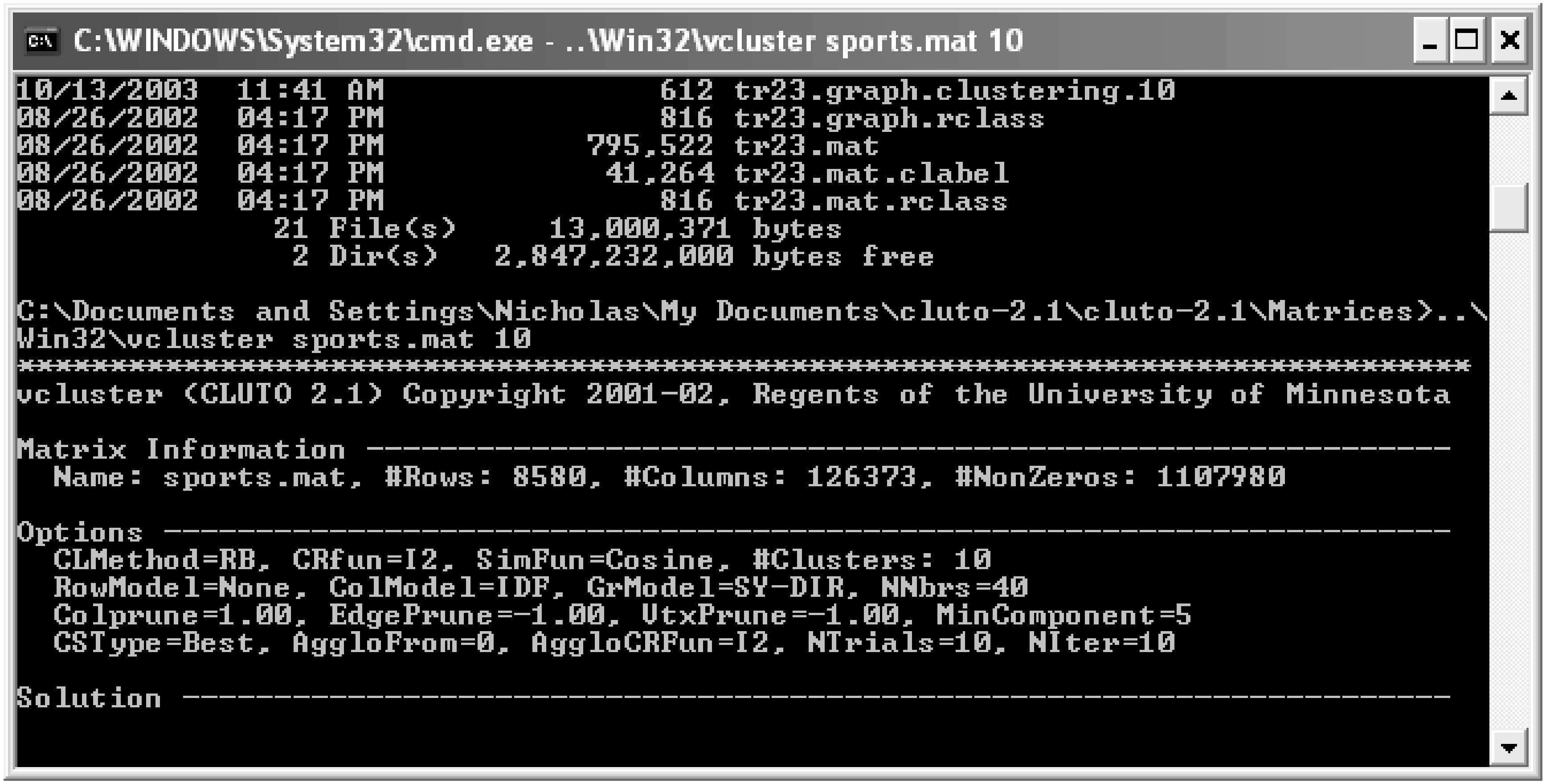
Means for ..	Protein
Cluster 1	1.75
Cluster 2	3.76
Cluster 3	7.12
Cluster 4	10.50
Cluster 5	10.15

ClustanGraphics File | 5 clusters | Click mouse to highlight cluster partition

Using Cluto

Need a Windows (or Unix) platform. Open a run window, run cmd.exe. From cluto’s matrices directory,

```
..\Win32\vcluster sports.mat 10
```



C:\WINDOWS\System32\cmd.exe - ..\Win32\vcluster sports.mat 10

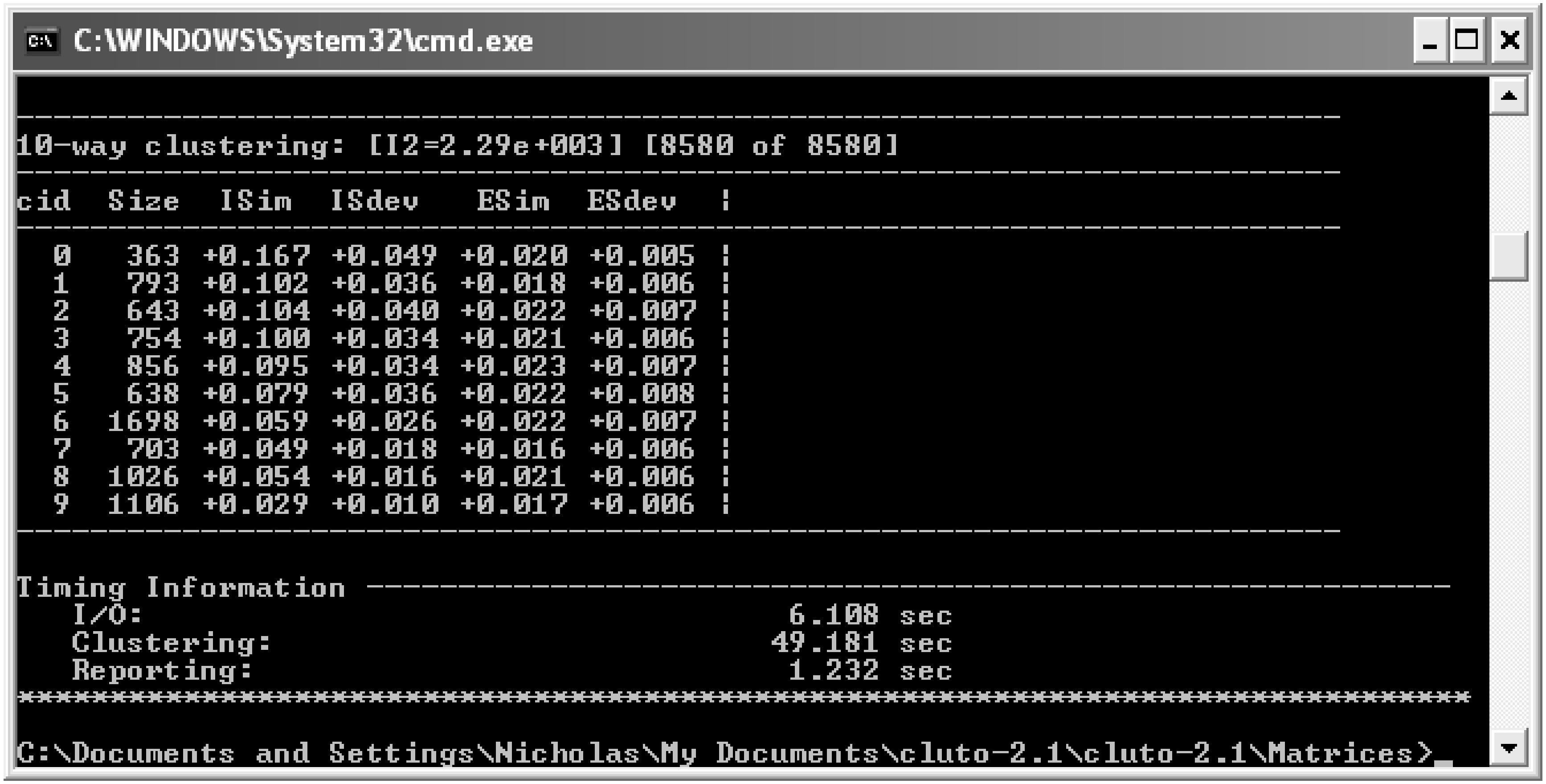
```
10/13/2003  11:41 AM               612 tr23.graph.clustering.10
08/26/2002  04:17 PM               816 tr23.graph.rclass
08/26/2002  04:17 PM           795,522 tr23.mat
08/26/2002  04:17 PM            41,264 tr23.mat.clabel
08/26/2002  04:17 PM               816 tr23.mat.rclass
              21 File(s)     13,000,371 bytes
               2 Dir(s)   2,847,232,000 bytes free

C:\Documents and Settings\Nicholas\My Documents\cluto-2.1\cluto-2.1\Matrices>..\
Win32\vcluster sports.mat 10
*******************************************************************************
vcluster (CLUTO 2.1) Copyright 2001-02, Regents of the University of Minnesota

Matrix Information -----------------------------------------------------------
  Name: sports.mat, #Rows: 8580, #Columns: 126373, #NonZeros: 1107980

Options ----------------------------------------------------------------------
  CLMethod=RB, CRfun=I2, SimFun=Cosine, #Clusters: 10
  RowModel=None, ColModel=IDF, GrModel=SY-DIR, NNbrs=40
  Colprune=1.00, EdgePrune=-1.00, VtxPrune=-1.00, MinComponent=5
  CSType=Best, AggloFrom=0, AggloCRFun=I2, NTrials=10, NIter=10

Solution ---------------------------------------------------------------------
```



```
------------------------------------------------------------------------
10-way clustering: [I2=2.29e+003] [8580 of 8580]
------------------------------------------------------------------------
cid  Size  ISim  ISdev   ESim  ESdev  |
------------------------------------------------------------------------
  0   363 +0.167 +0.049 +0.020 +0.005 |
  1   793 +0.102 +0.036 +0.018 +0.006 |
  2   643 +0.104 +0.040 +0.022 +0.007 |
  3   754 +0.100 +0.034 +0.021 +0.006 |
  4   856 +0.095 +0.034 +0.023 +0.007 |
  5   638 +0.079 +0.036 +0.022 +0.008 |
  6  1698 +0.059 +0.026 +0.022 +0.007 |
  7   703 +0.049 +0.018 +0.016 +0.006 |
  8  1026 +0.054 +0.016 +0.021 +0.006 |
  9  1106 +0.029 +0.010 +0.017 +0.006 |
------------------------------------------------------------------------

Timing Information -----------------------------------------------------------
   I/O:                                     6.108 sec
   Clustering:                             49.181 sec
   Reporting:                               1.232 sec
*******************************************************************************

C:\Documents and Settings\Nicholas\My Documents\cluto-2.1\cluto-2.1\Matrices>_
```

Related software

http://www.cs.umbc.edu/˜nicholas/clustering

http://www.cc.gatech.edu/˜dellaert/html/software.html

http://www.clustan.com/

http://www-users.cs.umn.edu/˜karypis/cluto/index.html

http://www.octave.org