Αναγνώριση Προτύπων

Ομαδοποίηση δεδομένων ΑΣΚΗΣΕΙΣ
(K-means and ISODATA Clustering)

Χριστόδουλος Χαμζάς

Τα περιεχόμενο αυτής της παρουσίασης βασίζεται στην παρουσίαση του αντίστοιχου διδακτέου μαθήματος του καθ. Kumar, University of Minessota. Βασίζεται στο Kef.8 &9 του βιβλίου: "Introduction to Data Mining", Chpt 8, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Addison-Wesley, 2006

Επιβλεπόμενη vs Μη-Επιβλεπόμενη Μάθηση

- Μέχρι τώρα θεωρήσαμε μεθόδους αναγνώρισης με classification όπου το πρότυπο χαρακτηρίζεται από τα μεγέθη {x,ω}
- Αυτά τα προβλήματα αναγνώρισης ονομάζονται Επιβλεπόμενα (supervised) αφού διατίθενται και το χαρακτηριστικό διάνυσμα και η σωστή απάντηση.
- Υπάρχουν όμως περιπτώσεις όπου δίνεται το χαρακτηριστικό διάνυσμα χωρίς την κλάση.
- Αυτές οι μέθοδοι καλούνται Μη-Επιβλεπόμενες (unsupervised) λόγω του ότι δεν χρησιμοποιούν τη σωστή απάντηση.

Επιβλεπόμενη vs Μη-Επιβλεπόμενη Μάθηση

Αν και οι μέθοδοι μη επιβλεπόμενης μάθηση φαίνονται περιορισμένων δυνατοτήτων υπάρχουν πολλές περιπτώσεις που επιβάλλεται η χρήση τους:

- Ο χαρακτηρισμός πολλών δεδομένων μπορεί να αποβεί δαπανηρός (π.χ. αναγνώριση ομιλίας)
- Το είδος της κλάσης μπορεί να μην είναι γνωστό εξ'αρχής.

Είδη αλγορίθμων ομαδοποίησης

- Ο K-means και οι παραλλαγές του
  - Παράγουν σετ ανεξάρτητων clusters
  - Οι πιο γνωστοί είναι οι k-means και ISODATA
- Ιεραρχικοί αλγόριθμοι
  - Το αποτέλεσμα είναι μια ιεραρχία εμφωλιασμένων clusters
  - Χωρίζονται στους ενωτικούς (agglomerative) και διαχωριστικούς (divisive)
- Πυκνωτικοί αλγόριθμοι

K-means Clustering

- Αλγόριθμος διαχωρισμού
- Κάθε ομάδα συνδέεται με ένα centroid (κέντρο)
- Κάθε σημείο αποδίδεται στην ομάδα με το πιο κοντινό κέντρο
- Ο αριθμός των ομάδων, K, **πρέπει να καθοριστεί**
- Πολύ απλός αλγόριθμος

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

Αξιολόγηση των ομάδων του K-means

- Η πιο κλασική μετρική είναι άθροισμα του τετραγώνου σφαλμάτων (Sum of Squared Error - SSE)
  - Για κάθε σημείο, το σφάλμα είναι η απόσταση από το κέντρο της κοντινότερης ομάδας
  - Το SSE είναι το άθροισμα του τετραγώνου σφαλμάτων αυτών:.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

όπου x είναι ένα σημείο στην ομάδα C_i και m_i είναι το αντιπροσωπευτικό σημείο (το κέντρο) για την ομάδα C_i

  - Δεδομένων δυο ομάδων, μπορούμε να επιλέξουμε αυτήν με το μικρότερο σφάλμα
  - Ένας εύκολος τρόπος για τη μείωση του SSE είναι η αύξηση του K, του αριθμού των ομάδων
    - Μια καλή ομαδοποίηση με μικρότερο K μπορεί να έχει μικρότερο SSE από μια κακή ομαδοποίηση με μεγαλύτερο K

K-means

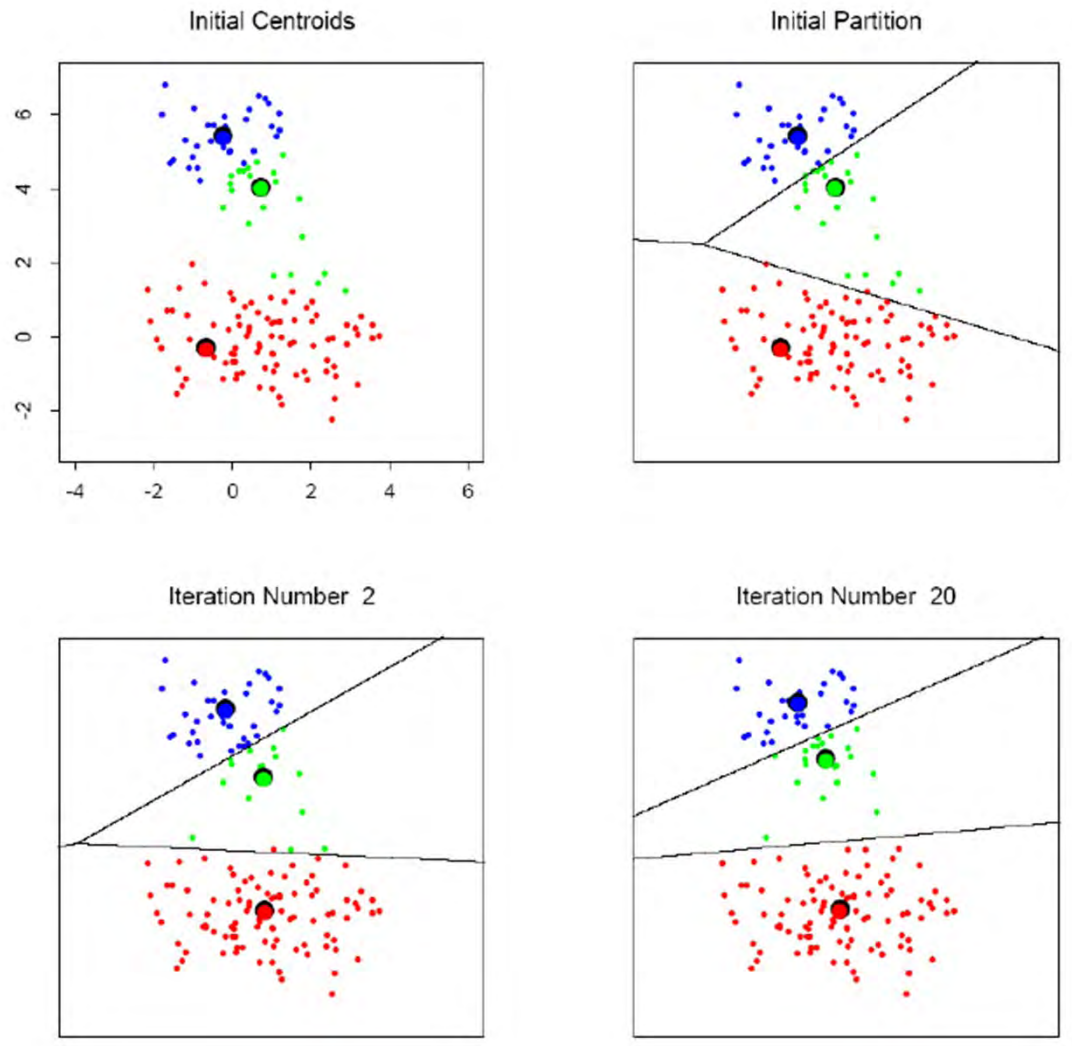
Το MatLab έχει μια δική του υλοποίηση του αλγορίθμου Kmeans.
Για λεπτομέρειες χρησιμοποίησε το **help kmeans**

Μία πιο απλή υλοποίηση είναι αυτή που δίνεται στο Πικράκη

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FUNCTION  [theta,bel,J]=k_means(X,theta)
% This function implements the k-means algorithm, which requires
% as input the number of clusters underlying the data set. The algorithm
% starts with an initial estimation of the cluster representatives and
% iteratively tries to move them into regions that are "dense" in data
% vectors, so that a suitable cost function is minimized. This is
% achieved by performing (usually) a few passes on the data set. The
% algorithm terminates when the values of the cluster representatives
% remain unaltered between two successive iterations.
%
% INPUT ARGUMENTS:
%  X:       lxN matrix, each column of which corresponds to
%           an l-dimensional data vector.
%  theta:   a matrix, whose columns contain the l-dimensional (mean)
%           representatives of the clusters.
%
% OUTPUT ARGUMENTS:
%  theta:   a matrix, whose columns contain the final estimations of
%           the representatives of the clusters.
%  bel:     N-dimensional vector, whose i-th element contains the
%           cluster label for the i-th data vector.
%  J:       the value of the cost function (sum of squared Euclidean
%           distances of each data vector from its closest parameter
%           vector) that corresponds to the  estimated clustering.
%
% (c) 2010 S. Theodoridis, A. Pikrakis, K. Koutroumbas, D. Cavouras
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [theta,bel,J]=k_means(X,theta)
[l,N]=size(X);
[l,m]=size(theta);
e=1;
iter=0;
while(e~=0)
    iter=iter+1;
    theta_old=theta;
    dist_all=[];
    for j=1:m
      dist=sum(((ones(N,1)*theta(:,j)'-X').^2)');
      dist_all=[dist_all; dist];
    end
% dist_all Nxm έχει τις αποστάσεις των N σημειων
από τα m κέντρα των clusters

    [q1,bel]=min(dist_all);
    J=sum(min(dist_all));

    for j=1:m
        if(sum(bel==j)~=0)

theta(:,j)=sum(X'.*((bel==j)'*ones(1,l))) /
sum(bel==j);
        end
    end
    e=sum(sum(abs(theta-theta_old)));
end
```

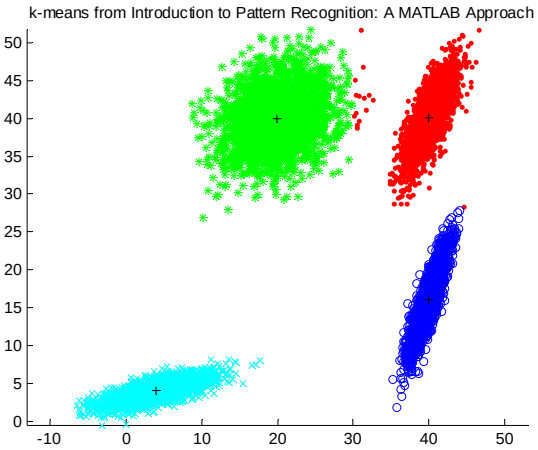
K-means clustering example



Παράδειγμα K-means

```
% Example example_Kmeans_ISODATA_cc
% for data clustering  C. Chamzas 15/6/2012

close('all');  clear; format compact ;

% 1. To generate the first N1 points of X1,

m=[1 1 ; 5 10 ; 10 4; 10 10]';
[l,c]=size(m);
S1=[0.8 0.2; 0.2 0.1]; S2=[0.8 0.2; 0.2 0.8];
S3=[0.1 0.25; 0.25 0.8]; S4=[0.2 0.3; 0.3 0.8];
S(:,:,1)=S1; S(:,:,2)=S2; S(:,:,3)=S3; S(:,:,4)=S4;
P=[1/4 1/4 1/4 1/4];

% 1. Generate X1 the data set
N1=10000;
randn('seed',0)
[X1,y1]=generate_gauss_classes(m,S,P,N1);
[l,N1]=size(X1);
X1=4*X1;

% Plot the data set
figure(1), hold on
plot(X1(1,:),X1(2,:),'.y')
figure(1), axis equal
```

```
% To apply the k-means algorithm for m = 4, work as in step
2 of Example 7.5.1. Pikrakis Attention data are given in
rows
m=4;  [l,N]=size(X1);
% places the initial centers randomly in the range of X1
theta_ini=rand(l,m);
X1_max=max(X1'); X1_min=min(X1');
for j=1:l
    theta_ini(j,:)=theta_ini(j,:)*(X1_max(j)-
X1_min(j))+X1_min(j);
end

[theta,bel,J]=k_means(X1,theta_ini);
SSE_pikrakis=J
theta
```



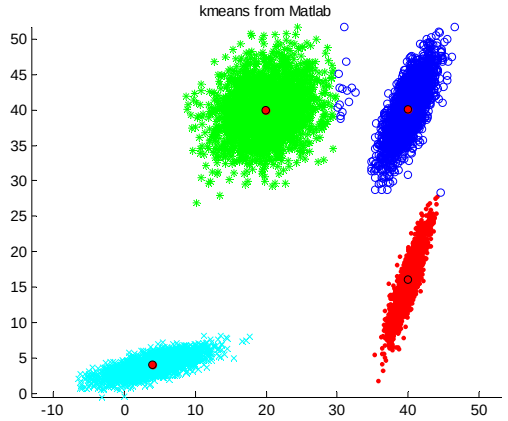
The 4 centers

40.01	19.94	39.97	3.96
16.00	39.92	40.08	3.97

SSE = 180940

```
……..
……..

%Matlab . Attention data are given in columns
K=4;
[IDX, C, SUMD] = kmeans(X1', K, 'Replicates',5);
SSE_matlab=sum(SUMD)
C=C'
```

```
% plots the segmented data
figure(3), hold on
figure(3),
plot(X1(1,IDX==1),X1(2,IDX==1),'r.',...
X1(1,IDX==2),X1(2,IDX==2),'g*',X1(1,IDX==3),X1(2,
IDX==3),'bo',...
X1(1,IDX==4),X1(2,IDX==4),'cx',X1(1,IDX==5),X1(2,
IDX==5),'md',...
X1(1,IDX==6),X1(2,IDX==6),'yp',X1(1,IDX==7),X1(2,
IDX==7),'ks')
figure(3),
plot(C(1,:),C(2,:),'ko','MarkerFaceColor','r')
title('kmeans from Matlab');
figure(3), axis equal
```



The 4 centers

40.01	19.94	39.97	3.96
16.00	39.92	40.08	3.97

SSE = 180940

17 iterations, total sum of distances = 180940
14 iterations, total sum of distances = 659350
4 iterations, total sum of distances = 180940
4 iterations, total sum of distances = 180940
4 iterations, total sum of distances = 180940

```
opts = statset('Display','final');
[IDX, C, SUMD] = kmeans(X1', K, 'Replicates',5, …
'Replicates',5, 'Options',opts);
```

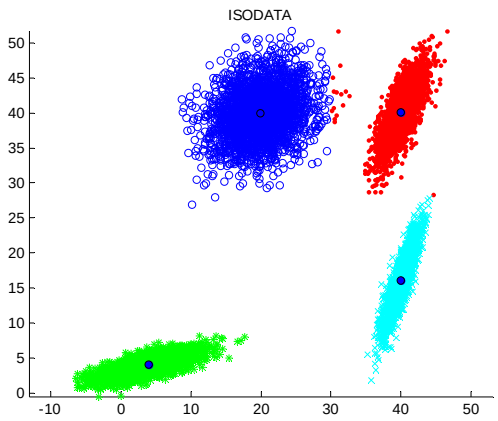
ISODATA

- ISODATA είναι συντομογραφία του Iterative Self-Organizing Data Analysis Technique Algorithm
- Είναι επέκταση του k-means που εμπεριέχει ευριστικούς τρόπους για την αυτόματη επιλογή του πλήθους των κλάσεων
- Ο χρήστης επιλέγει τις παραμέτρους (Παράμετροι προγράμματος):
  - NMIN_EX ελάχιστο πλήθος δειγμάτων ανά cluster (ON)
  - ND επιθυμητό μέγιστο πλήθος cluster (K)
  - σ_s^2 μέγιστη διασπορά για διαχωρισμό clusters (OS)
  - DMERGE μέγιστη απόσταση για ένωση clusters (OC)
  - NMERGE μέγιστο πλήθος clusters που μπορούν να ενωθούν (L)

How ISODATA Works:

1. Cluster centers are randomly placed and pixels are assigned based on the shortest distance to center method
2. The standard deviation within each cluster, and the distance between cluster centers is calculated
   - Clusters are split if one or more standard deviation is greater than the user-defined threshold
   - Clusters are merged if the distance between them is less than the user-defined threshold
3. A second iteration is performed with the new cluster centers
4. Further iterations are performed until:
   - the average inter-center distance falls below the user-defined threshold,
   - the average change in the inter-center distance between iterations is less than a threshold, or
   - the maximum number of iterations is reached

```
%ISODATA
ON=15; % threshold number of elements for the elimination of a cluster.
OC=5; % threshold distance for the union of clusters.
OS=7;  % deviation typical threshold for the division of a cluster.
k=7;   % number (maximum) cluster.
L=2;   % maximum number of clusters that can be mixed in a single iteration.
I=10;  % maximum number of iterations allowed.
NO=1;  % extra parameter to automatically answer no to the request of cambial any parameter.
min_dist=50; % Minimum distance a point must be in each center. If you want to delete any point
       % Give a high value.
%%%%%%%%%%%%%%%%%%%%%
%  Function ISODATA  %
%%%%%%%%%%%%%%%%%%%%%
[C_ISO, Xcluster, A, clustering]=isodata_ND(X1', k, L, I, ON, OC, OS, NO, min_dist);
C_ISO=C_ISO'; IDX_ISO=clustering';
% Evaluation the SSE error
[l,m]=size(C_ISO);
dist_all=[];
for j=1:m
 dist=sum(((ones(N,1)*C_ISO(:,j)'-X1').^2)');
 dist_all=[dist_all; dist];
end
SSE_ISODATA=sum(min(dist_all))
fprintf('Number of Clusters: %d\n',A);
C_ISO
```

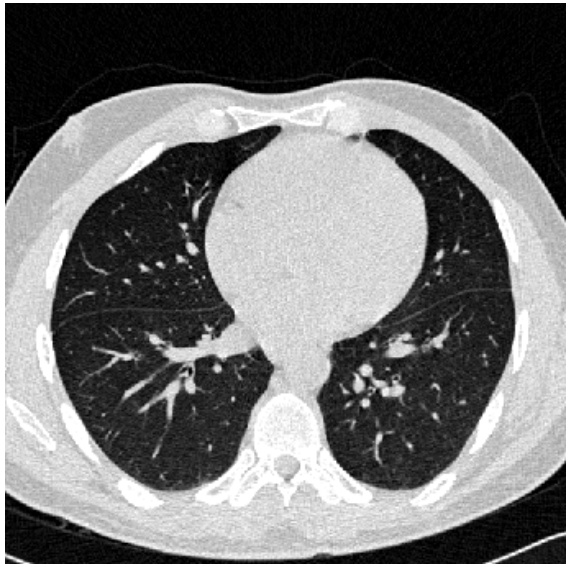


The 4 centers

40.01	19.94	39.97	3.96
16.00	39.92	40.08	3.97

SSE = 180940

K-means Image Segmentation



An image (*I*)

Three-cluster image (*J*) on gray values of *I*

Matlab code:

```
I = double(imread('...'));

J = reshape(kmeans(I(:),3),size(I));
```

Note that *K*-means result is "noisy"

Παράδειγματα για Εικόνες

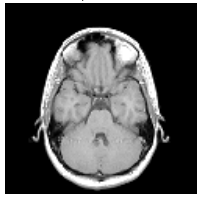
```
% Example "example_Kmeans_Images_cc.m"
% kmeans for image segnentation
% C. Chamzas 15/6/2012

close('all');  clear; format compact ;

Im = imread('trees.tif');
imshow(Im);
I = double(Im);
```

```
% segments it in 3 gray level colors
[IDX, C, SUMD]=kmeans(I(:),3,'Replicates',5);
J = reshape(IDX,size(I));
mapGRAY=[C';C';C']'/max(I(:));
figure(2); imshow(J,mapGRAY);
```

```
% segments it in 5 gray level colors
[IDX, C, SUMD]=kmeans(I(:),5,'Replicates',5);
J = reshape(IDX,size(I));
mapGRAY=[C';C';C']'/max(I(:));
figure(3); imshow(J,mapGRAY);
```

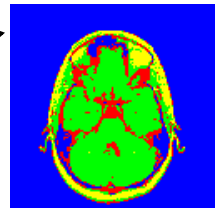
```
load mri
I = D(:,:,8);
figure(6); imshow(I,map);
title('MRI image_num = 8');
axis image
```

MRI image num = 8



```
% segments it in 4  level colors
I=double(I);
[IDX, C,
SUMD]=kmeans(I(:),4,'Start','uniform','Replicates',5
,'EmptyAction','singleton');
J = reshape(IDX,size(I));
mapRGB=[ 1 0 0; 0 1 0; 0 0 1; 1 1 0; 1 0 1; 0 1 1];
mapGRAY=[C';C';C']'/max(I(:));
figure(7); imshow(J,mapRGB); title('4 colors');
figure(8); imshow(J,mapGRAY);title('4 gray levels');
```

4 colors



4 gray levels