Machine Learning and Data Mining
Clustering

Patricia Hoffman, PhD

Clustering
Main Sources

- K-Means (Andrew Ng)
  - http://cs229.stanford.edu/notes/cs229-notes7a.pdf
- Expectation – Maximization (Andrew Ng)
  - http://www.youtube.com/watch?v=ey2PE5xi9-A&feature=related
  - http://cs229.stanford.edu/notes/cs229-notes2.pdf
- EM Algorithm (Andrew Ng)
  - http://cs229.stanford.edu/notes/cs229-notes8.pdf
  - http://cs229.stanford.edu/notes/cs229-notes7b.pdf
- Discriminant Analysis (Andrew Ng)
  - http://cs229.stanford.edu/notes/cs229-notes2.pdf
- R and Expectation Maximization
  - http://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Expectation_Maximization_%28EM%29
- Paper Describing mclust package
  - http://www.stat.washington.edu/fraley/mclust/tr504.pdf

R Packages

- CRAN list of Cluster Packages
  - http://cran.r-project.org/web/views/Cluster.html
- CRAN K-Means kmeans()
  - http://127.0.0.1:57279/rhelp/browse/user-local--1vj6d1s7n4kx5/library/stats/html/kmeans.html
- CRAN Agglomerative Hierarchal Clustering
  - stat package hclust() Installed with R
- CRAN Divisive Hierarchal Clustering diana()
  - http://cran.r-project.org/web/packages/cluster/cluster.pdf
- CRAN Mixture Model Package mclust()
  - http://cran.r-project.org/web/packages/mclust/mclust.pdf
  - http://www.stat.washington.edu/research/reports/2006/tr504.pdf
- CRAN Density Based Clustering dbscan()
  - http://cran.r-project.org/web/packages/fpc/index.html

Cluster Analysis

- Huge variety in R

http://cran.r-project.org/web/views/Cluster.html

- Methods
  - Partitioning (e.g., kmeans, pam)
  - Hierarchical Agglomerative (e.g., average, ward, single, complete) (hclust() and diana())
  - Model Based (e.g., ML estimation, Bayesian estimation) (mclust())
  - Density Based (e.g., ML estimation, Bayesian estimation) (dbscan()) page 35 of fpc package

Example Code

The Iris Data is globular with ellipsoidal covariance

- kMeans.R kmeans() Prototype Based Clustering
  - Randomly Generated Gaussian Data
  - Iris Data Set 89% correct
- hclustExample.R Hierarchical Clustering
  - USArrests Data
  - Iris Data Set
    - hclust() Agglomerative Hierarchical 91% correct
    - diana() Divisive Hierarchical 85% correct
- mclustExample mclust() Expection - Maximization
  - Iris Data 97% Correct

Lessons

- Measures.pdf
- Chap8_basic_cluster_analysis.pdf (kmeans)
- Video of kmeans
  - http://www.youtube.com/watch?v=74rv4snLl70&feature=endscreen&NR=1
- Example Code Kmeans.r
- Chap8_basic_cluster_analysis.pdf (Hierarchical Clustering)
- Expectation – Maximization
- Video of Expectation – Maximization
  - http://www.youtube.com/watch?v=v-pq8VCQk4M&feature=related
- Example Code (mclustExample.r)

Classes vs. Clusters

- **Supervised**: $X = \{ \boldsymbol{x}j, \boldsymbol{y}j \}_t$
- Classes C_i $i=1,...,K$

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x} \mid C_i) P(C_i)$$

where $p(\boldsymbol{x} \mid C_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- $\Phi = \{P(C_i), \boldsymbol{\mu}_i, \Sigma_i\}^K_{i=1}$

$$\hat{P}(C_i) = \frac{\sum_j yj_i}{N} \quad \mathbf{m}_i = \frac{\sum_j yj_i \, \mathbf{x}j}{\sum_j yj_i}$$

$$\mathbf{S}_i = \frac{\sum_j yj_i (\mathbf{x}j - \mathbf{m}_i)(\mathbf{x}j - \mathbf{m}_i)^T}{\sum_j yj_i}$$

Unsupervised: $X = \{ \boldsymbol{x}j \} j$

- Clusters G_i $i=1,...,k$

$$p(\boldsymbol{x}) = \sum_{i=1}^{k} p(\boldsymbol{x} \mid G_i) P(G_i)$$

where $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \Sigma_i)$

- $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \Sigma_i\}^k_{i=1}$

No target Labels, *y j*

(where the covariance matrix = Σ_i)

EM - Clusters

Unsupervised: X = { $\boldsymbol{x}^j$ }$_j$

- Clusters G_i i=1,...,k

$$p(\mathbf{x}) = \sum_{i=1}^{k} p(\mathbf{x} \mid G_i) P(G_i)$$

where $p(\boldsymbol{x} \mid G_i) \sim N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$

- $\Phi = \{P(G_i), \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{k}$

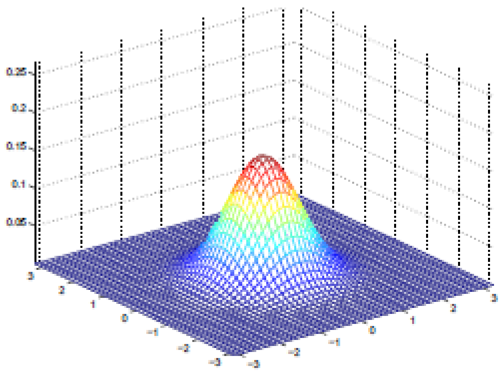
No target Labels, *y j*

(where the covariance matrix = $\boldsymbol{\Sigma}_i$)

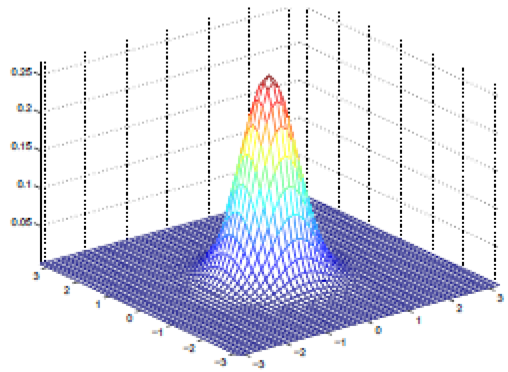
- G_i are the mixture components – group or clusters
- $P(G_i)$ are the mixture proportions
- k is the number of components
  - specified beforehand
- $p(\boldsymbol{x} \mid G_i)$ and Φ are the parameters that should be estimated from the iid sample X = { $\boldsymbol{x}^j$ }$_j$

Goal: Find the component density parameters that maximize the likelihood of the sample. That is, find the parameter vector Φ that maximizes the likelihood of the observed values of X = { $\boldsymbol{x}^j$ }$_j$

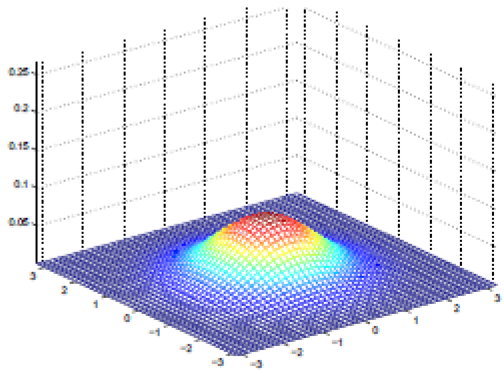
Gaussian Distributions



$\mu = 0; \Sigma = I$



$\mu = 0; \Sigma = (0.6)\, I$

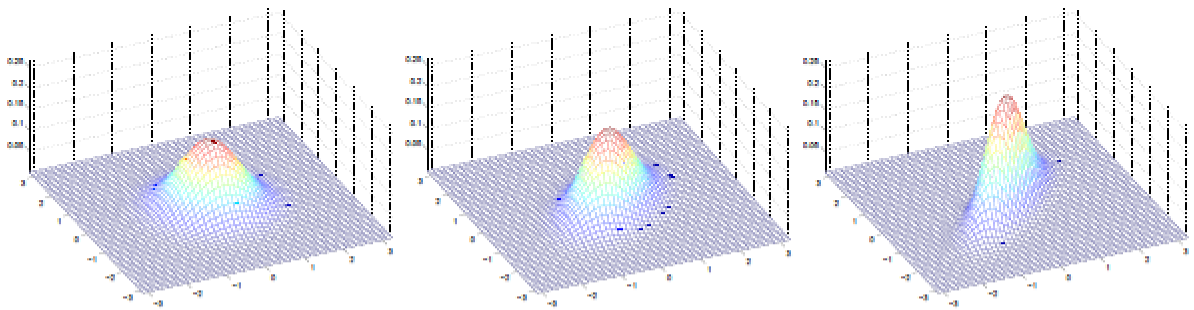


$\mu = 0; \Sigma = 2I$

(I = 2x2 identity matrix)

http://cs229.stanford.edu/notes/cs229-notes2.pdf

Gaussians with mean 0



$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Covariance Matrices as indicated

http://cs229.stanford.edu/notes/cs229-notes2.pdf

Covariance Matrix Σ

- Let X be a set of n random variables
  - $X = (X1, X2, \ldots, Xn)^T$
- Covariance between

  Xi and Xj = $E[(Xi - \mu i)(Xj - \mu j)]$
- The ijth entry of the covariance matrix

  $\Sigma ij = cov(Xi,Xj) = E[(Xi - \mu i)(Xj - \mu j)]$
- Note that Σii is the variance of Xi
- $\Sigma = E[(X - E[X])(X - E[X])^T]$

Gaussian Mixture Model

- Each xj is generated randomly
  - choose I {1 to k} to select Gi, one of the Gaussians
  - Use the parameters of Gi to generate xj
- So, assuming your data was generated as a Gaussian Mixture Model, the Expectation – Maximization Algorithm can be used with Gaussians
- Note: other distributions can be used
  - Usually z is used to denote the hidden target (unsupervised)
    - Ie zi = Gi
  - For supervised methods, target was denoted by y

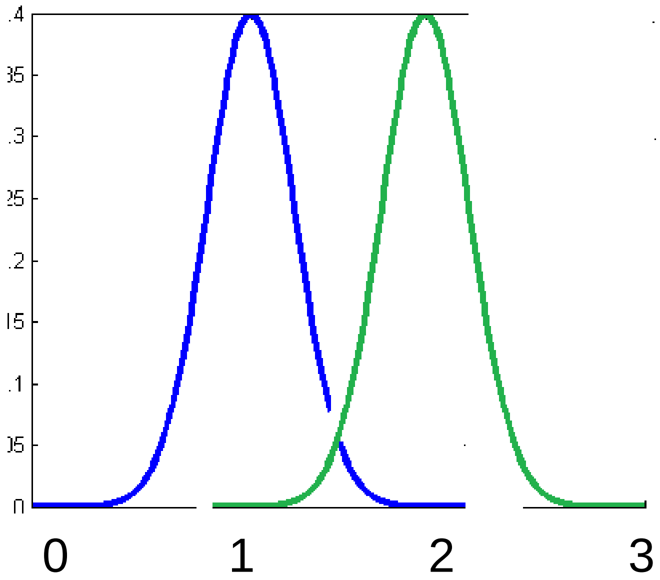
Mixture of Mixtures

- In classification, the input comes from a mixture of classes (supervised).
- If each class is also a mixture, e.g., of Gaussians, (unsupervised), we have a mixture of mixtures:

$$p(\boldsymbol{x} \mid C_i) = \sum_{j=1}^{k_i} p(\boldsymbol{x} \mid G_{ij}) P(G_{ij})$$

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} p(\boldsymbol{x} \mid C_i) P(C_i)$$

Example – 2 Gaussian Distributions



- Classify Data less than 1.5 as Blue
- Classify Data greater than 1.5 as Green

P(x = 0.5 ∈ Φblue) > P(x = 0.5 ∈ Φgreen) so classify x = 0.5 as blue

Assume there are two Classes, each with a Gaussian Distribution with know mean and variance. Classify a given observation by choosing the Gaussian distribution which maximizes the probability .

Intuition for Gaussian Distribution

(Where does Maximization Step Come From?)

Expectation: Given μ and σ the probability distribution function for a point x:

$$\text{Probability Distribution Function} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2} \right]$$

Maximization: Given a set of points x_j,

$$\text{mean } \mu = \sum_{j=1}^{k} x_j$$

$$\text{standard deviation } \sigma = \left(\sum_{j=1}^{k} (x_j - \mu)^2 \right)^{1/2}$$

Maximum Likelihood Derivation of mean and std for Gaussian Distribution

- Maximize the Likelihood function:

$$l(\phi) = p(\mathbf{X}|\phi) = \prod_{j=1}^{N} p(xj|\phi)$$

- Easier to Maximize the Log of the Likelihood function:

$$L(\phi|\mathbf{X}) = \log l(\phi|\mathbf{X}) = \sum_{j=1}^{N} \log p(xj|\phi)$$

- For Gaussian Distributions:

$$L(\mu,\sigma|\mathbf{X}) = -\frac{N}{2}\log(2\pi) - N log\sigma - \frac{\sum_j (xj-\mu)^2}{2\sigma^2}$$

Maximum Likelihood Estimation

- Suppose X = $\{xj\}_{j=1}^{N}$ where xj are independent identically distributed (iid) samples from some known probability density family, p(x| Φ) defined up to parameters, Φ : xj ~ p(x| Φ)
- Find Φ that makes sampling xj from p(x| Φ) as likely as possible.
- The "likelihood" of the sample X given the parameter Φ is the product of the likelihoods of the individual points:

$$l(\phi) = p(\mathbf{X}|\phi) = \prod_{j=1}^{N} p(xj|\phi)$$

(note throughout Φ is the same as ϕ)

Maximum Likelihood Derivation
of mean for Gaussian Distribution

- Holding X fixed, maximize the likelihood =

  max$_{\mu,\sigma}$($L(\mu, \sigma|\mathbf{X}) = -\frac{N}{2}\log(2\pi) - N log \sigma - \frac{\sum_j (xj-\mu)^2}{2\sigma^2}$)

- Set the Derivative with respect to μ equal to 0

  and solve for μ:

$$\frac{\partial L}{\partial \mu} = 0 \Rightarrow \quad \sum_j xj = \sum_{j=1}^{N} \mu \Rightarrow \quad \sum_j xj = N\mu \Rightarrow$$

$$\mu = \frac{\sum_j xj}{N}$$

Similar derivation for Standard Deviation

Clustering Using Mixture Models

- Assume the data belongs to a set of specific distributions {Gi} for i=1 to k
  - Example: Gaussians
- Each distribution corresponds to a cluster
- The parameters of each distribution provide a description of the corresponding cluster
- Estimate the Parameters Φi for each Distribution
  - Gaussian – mean & standard deviation ie $\Phi i = \{\mu i, \sigma i\}$
- Classify an object by putting it in the cluster for which it has the maximum probability

Expectation Maximization Algorithm

Initialize the set of model parameters

For Gaussian Class Gi, $\Phi i = \{\mu i, \sigma i\}$ for i = 1 to K

Repeat:

Expectation Step: For each xj in the data set, calculate the probability that xj belongs to Class Gi, i.e., For each i calculate: $P(xj \in \text{Class } Gi \mid \Phi i)$

Maximization Step: Given the probabilities from the expectation step, find the new estimates of the parameters Φi that maximize the expected likelihood for the data set.

Until: The change in parameters, Φi , is below a set threshold.

Expectation Maximization Details

Expectation Step: wij = p(xj ∈ Class Gi | Φi)

wij is the probability that observation xj is in class Gi

Maximization Step:

$$wi := \frac{1}{m}\sum_{j=1}^{N} wij$$

$$\mu_i := \frac{\sum_{j=1}^{N} wijxj}{\sum_{j=1}^{N} wij}$$

$$\text{the covarience matrix}\left(\sum i\right) := \frac{\sum_{j=1}^{N} wij(xj-\mu_i)(xj-\mu_i)^T}{\sum_{j=1}^{N} wij}$$

Bayesian Information Criterion (BIC)

- BIC = maximized log of likelihood with penalty for number of parameters in model.
- BIC allows comparison of models with differing parameterizations and/or differing number of clusters.
- BIC = $-2\ln L + m\ln(N)$ where
  - L = Maximized value of the likelihood
  - m = number of free parameters to estimate
  - N = the number of observations
- Lower BIC is better – requires fewer explanatory variables

Iris Data – BIC chooses VEV

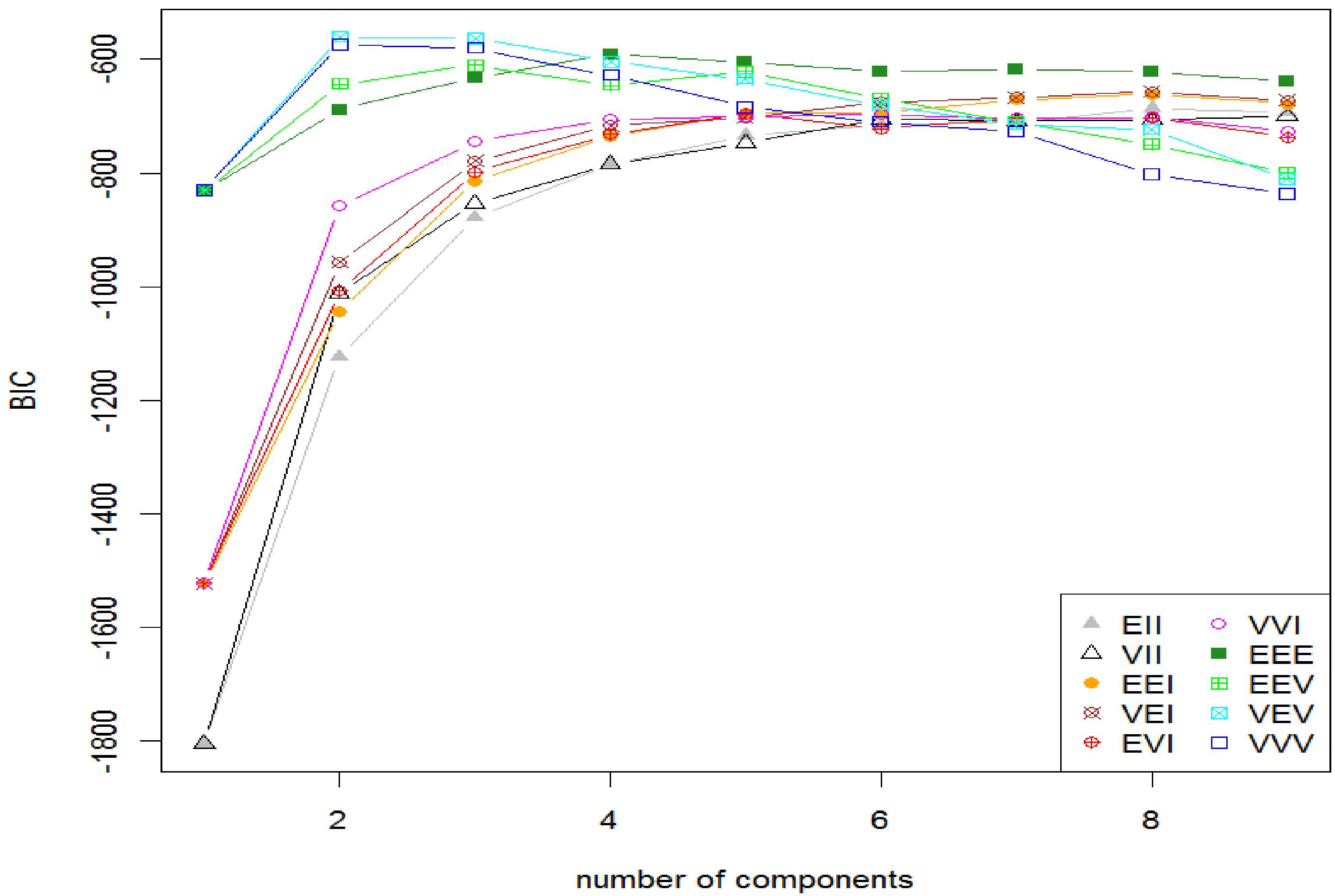
Parameterizations of the covariance matrix Σ currently available in MCLUST for hierarchical clustering (HC) and/or EM for multidimensional data.

(• indicates availability).

identifier	Model	HC	EM	Distribution	Volume	Shape	Orientation
E		•	•	(univariate)	equal		
V		•	•	(univariate)	variable		
EII	λI	•	•	Spherical	equal	equal	NA
VII	$\lambda_k I$	•	•	Spherical	variable	equal	NA
EEI	λA		•	Diagonal	equal	equal	coordinate axes
VEI	$\lambda_k A$		•	Diagonal	variable	equal	coordinate axes
EVI	λA_k		•	Diagonal	equal	variable	coordinate axes
VVI	$\lambda_k A_k$		•	Diagonal	variable	variable	coordinate axes
EEE	$\lambda D A D^T$	•	•	Ellipsoidal	equal	equal	equal
EEV	$\lambda D_k A D_k^T$		•	Ellipsoidal	equal	equal	variable
VEV	$\lambda_k D_k A D_k^T$		•	Ellipsoidal	variable	equal	variable
VVV	$\lambda_k D_k A_k D_k^T$	•	•	Ellipsoidal	variable	variable	variable

http://www.stat.washington.edu/research/reports/2006/tr504.pdf

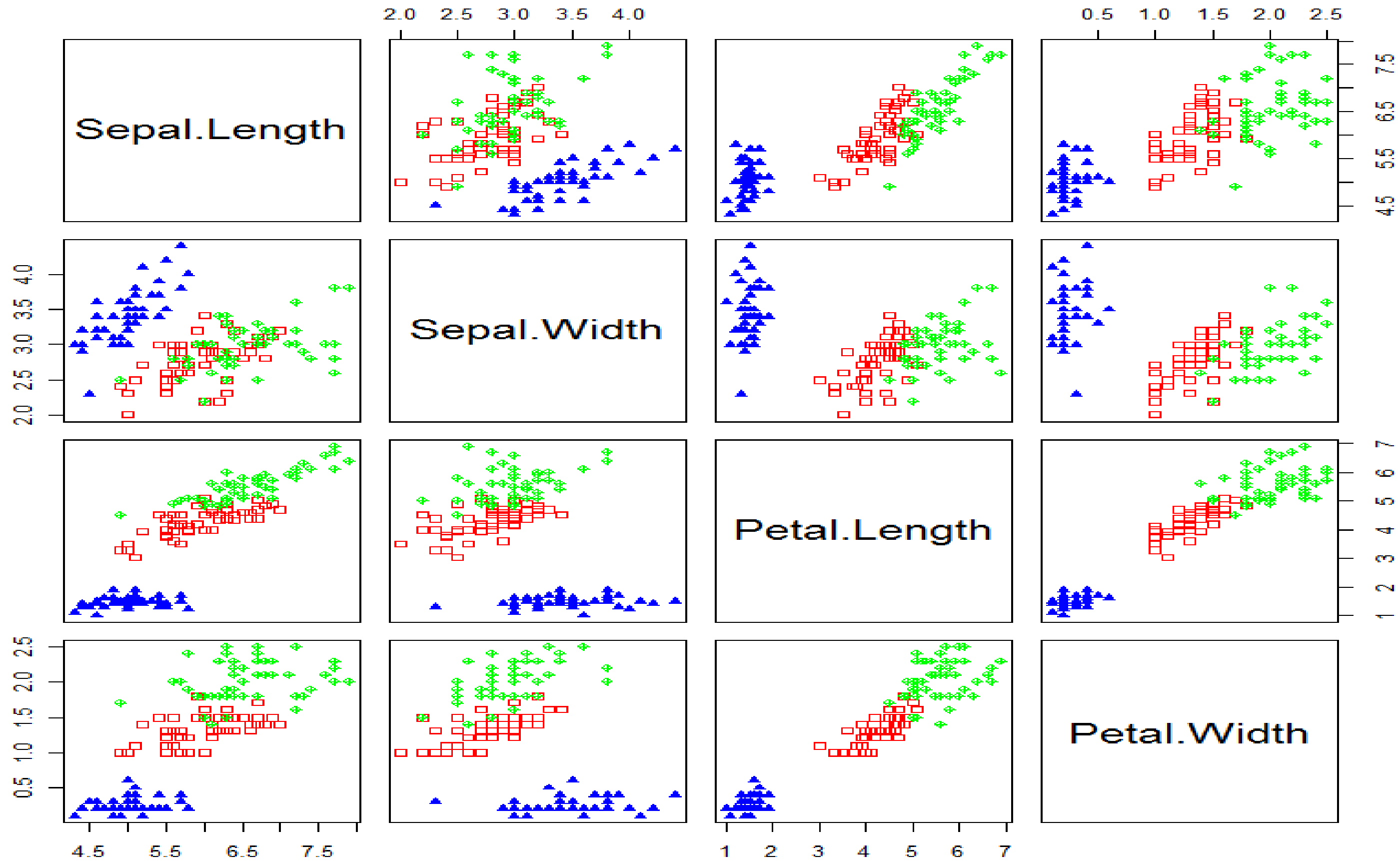
BIC Plot for Iris Data (VEV)



Iris Data and BIC

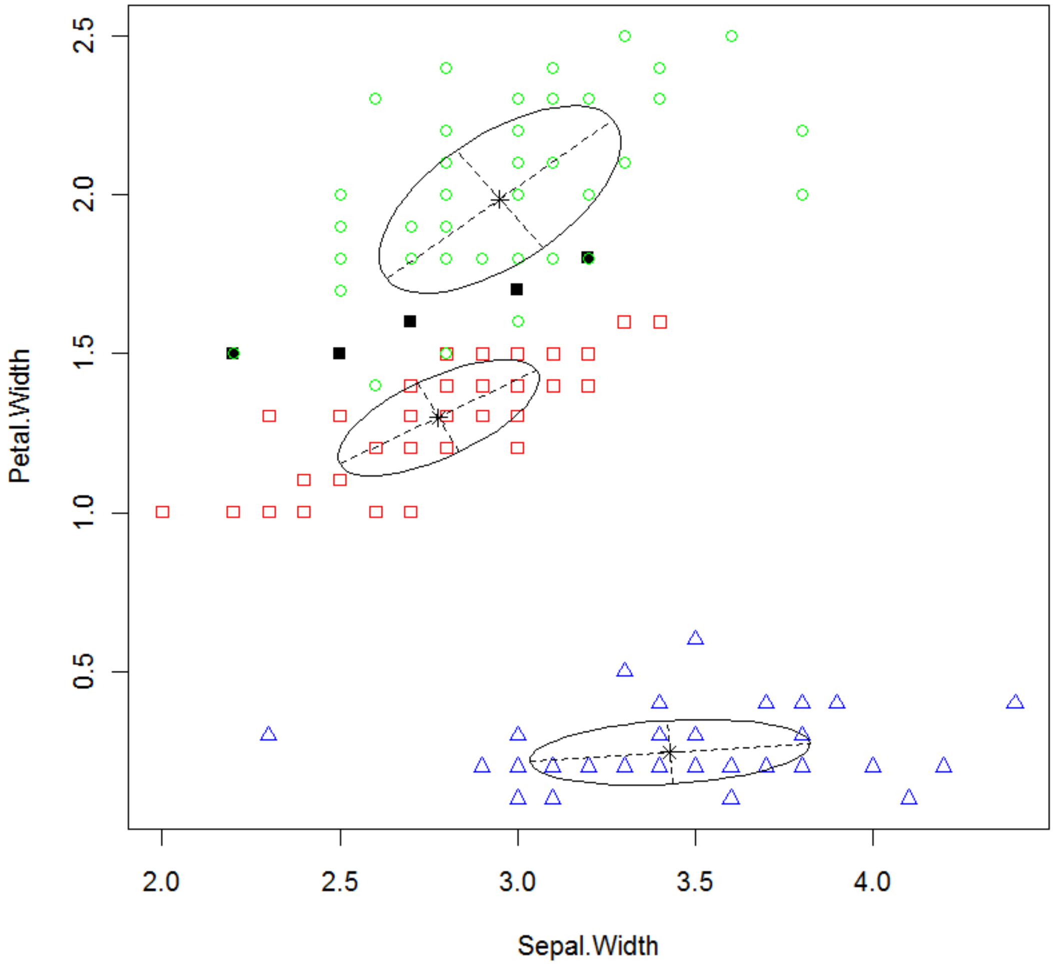
- Best Model for Iris Data is VEV using 2 clusters
  - mclustBIC(iris[,-5])
- However Iris Data has 3 classes
  - Must force 3 clusters
    - mclustBIC(iris[,-5],G = 3)

Pairs Plot for Iris Data



Iris Data

Notice that covariance are ellipsoid not oriented with axis



VEV – volume and orientation are variable

classification table:

1	2	3
50	45	55

EM characteristics

- Objects being classified must be elements of a finite dimensional vector-space over the real numbers
- Wide variety of shapes (long skinny clusters, clusters with holes punched in the middle, etc.)
- Less sensitive to large density variations than k-means
- Generative model
  - maximum likelihood parameters for a probabilistic model

K- Means Characteristics

- Works for any data for which you can define a distance or a similarity measure
- Roughly spherical (convex polyhedral) and roughly same volume

After Clustering

- Dimensionality reduction methods find correlations between features and group features
- Clustering methods find similarities between instances and group instances
- Allows knowledge extraction through

  number of clusters,

  prior probabilities,

  cluster parameters, i.e., center, range of features.

  Example: CRM, customer segmentation