International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 12 150 – 155

Performance Analysis of a Gaussian Mixture based Feature Selection Algorithm

B. V. Swathi
Professor, Dept. of CSE
Geetanjali College of Engg.& Tech
Keesara,India
e-mail: swathiveldanda@yahoo.com

***Abstract*—**Feature selection for clustering is difficult because, unlike in supervised learning, there are no class labels for the data and, thus, no obvious criteria to guide the search. The work reported in this paper includes the implementation of unsupervised feature saliency algorithm (UFSA) for ranking different features. This algorithm used the concept of feature saliency and expectation-maximization (EM) algorithm to estimate it, in the context of mixture-based clustering. In addition to feature ranking, the algorithm returns an effective model for the given dataset. The results (ranks) obtained from UFSA have been compared with the ranks obtained by Relief-F and Representation Entropy, using four clustering techniques EM, Simple K-Means, Farthest-First and Cobweb.For the experimental study, benchmark datasets from the UCI Machine Learning Repository have been used.

Keywords*-*gaussian mixtures,clustering,unsupervised,feature selection,relief-F

*****

I. INTRODUCTION

In machine learning, feature selection, also known as variable selection, feature reduction, attribute selection or variable subset selection, is the technique of selection a subset of relevant features for building robust learning models.

Feature selection is a must for any data mining product. That is because, when you build a data mining model, the dataset frequently contains more information than is needed to build a model. For example, a dataset may contain 500 columns that describe characteristics of customers, but perhaps only 50 of those columns are used to build a particular model. If you keep the unneeded columns while building the model, the clusters will not be well defined and more storage space is required for the completed model.

Feature selection[5] works by calculating a score for each attribute, and then selecting only the attributes that have the best scores. You can adjust the threshold for the top scores. Feature selection is always performed before the model is trained, to automatically choose the attributes in a dataset that are most likely to be used in the model.

There are various methods for feature selection. The exact method for selecting the attributes with the highest value depends on the algorithm used in your model, and any parameters that you may have set on your model. Feature selection is applied to inputs, predictable attributes, or to states in a column. Only the attributes and states that the algorithm selects are included in the model-building process and can be used for prediction. Predictable columns that are ignored by feature selection are used for prediction, but the predictions are based only on the global statistics that exist in the model.

.

II. BACKGROUND

In statistics, a Mixture Model is a probabilistic model for representing the presence of sub-populations within an overall population. This model does not require that an observed data-set should identify the sub-population to which an individual observation belongs.

Formally a mixture model corresponds to the mixture distribution that represents probability distribution of observations in the overall population. However, while problems associated with "mixture distributions" relate to deriving the properties of the overall population from those of the sub-populations, "mixture models" are used to make statistical inferences about the properties of the sub-populations given only observations on the pooled population, without sub-population-identity information.

The methods which can be used to implement such mixture models[1] can be called as unsupervised learning or clustering methods.

A. General mixture model

A typical finite-dimensional mixture model is a hierarichal model consisting of the following components:

- N random variables corresponding to observations, each assumed to be distributed according to a mixture of K components, with each component belonging to the same parametric family of distributions (eg, all Normal) but with different parameters.
- N corresponding random latent variables specifying the identity of the mixture component of each observation, each distributed according to a D-dimensional categorical distribution.
- A set of L mixture weights, each of which is a probability (a real number between 0 and 1), all of which sum to 1.
- A set of L parameters, each specifying the parameter of the corresponding mixture component. In many cases, each "parameter" is actually a set of parameters. For example, observations distributed according to a mixture of one-dimensional Gaussian distribution will have a mean and variance for each component. Observations distributed according to a mixture of D-dimensional categorical distributions (e.g., when each observation is a word from a vocabulary of size D) will have a vector of D probabilities, collectively summing to 1).

The common possibilities for the distribution of the mixture components are:

- Binomial distribution, for the number of "positive occurrences" (e.g., successes, yes votes, etc.) given a fixed number of total occurrences
- Multinomial distribution, similar to the binomial distribution, but for counts of multi-way occurrences (e.g., yes/no/maybe in a survey)
- Negative binomial distribution, for binomial-type observations but where the quantity of interest is the number of failures before a given number of successes occurs.
- Poisson distribution, for the number of occurrences of an event in a given period of time, for an event that is characterized by a fixed rate of occurrence.
- Exponential distribution, for the time before the next event occurs, for an event that is characterized by a fixed rate of occurrence.
- Log-normal distribution, for positive real numbers that are assumed to grow exponentially, such as incomes or prices.
- Multivariate normal distribution (multivariate Gaussian distribution), for vectors of correlated outcomes that are individually Gaussian-distributed..

B. Model based Clustering

In this type of clustering, certain models for clusters are used and we attempt to optimize the fit between the data and the model. In practice, each cluster can be mathematically represented by a parametric distribution, like a Gaussian (continuous) or a Poisson (discrete). The entire data set is therefore modeled by a *mixture* of these distributions. An individual distribution used to model a specific cluster is often referred to as a *component*distribution.

A mixture model with high likelihood tends to have the following traits:

- component distributions have high “peaks” (data in one cluster are tight);
- the mixture model “covers” the data well (dominant patterns in the data are captured by component distributions).

The most widely used clustering method of this kind is the one based on learning a mixture of Gaussians: we can actually consider clusters as Gaussian distributions centered on their centers.

Main advantages of model-based clustering:

- well-studied statistical inference techniques available;
- flexibility in choosing the component distribution;
- obtain a density estimation for each cluster
- a “soft” classification is available.

C. Gaussian Mixture Model

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. These are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system, such as vocal-tract related spectral features in a speaker recognition system.

GMM parameters[5] are estimated from training data using the iterative Expectation Maximisation algorithm. The complete Gaussian Mixture model is parameterized by the mean vectors, covariance vectors and mixture weights from all component densities.

It is also important to note that because the component Gaussians are acting together to model the overall feature density, full covariance matrices are not necessary if the features are not statistically independent. The linear combination of diagonal covariance basis Gaussians is capable of modelling the correlations between feature vector elements.

III. UNSUPERVISED FEATURE SALIENCY ALGORITHM FORFEATURE SELECTION (UFSA)

The feature selection algorithm has been implemented with unsupervised feature saliency approach based on the paper in [1]. This algorithm is an extension of an EM algorithm for performing mixture based clustering with feature selection. The algorithm gives various models for given data set and feature saliency values for each feature in a given model. Feature saliency becomes zero if a feature is irrelevant. Based on feature saliency of features in each iteration, rank is assigned to each feature (implies feature saliency is considered as a metric for feature ranking). Also among different models an effective model can be returned based on the minimum message length criterion.

Unsupervised Feature Saliency algorithm involves

1. Initialization Step.
2. Expectation Step.
3. Maximization Step.
4. Feature ranking.
5. Recording the model and its message length.
6. Final step

A. Intialization Step

In this step we initialize all the parameters of the mixture components and the common distribution (which covers all the data points), initial number of components and feature saliencies of all the features. The parameters for defining Gaussian distribution are mean and variance.

Step1: Randomly initialize the parameters for large number of mixture components. These initial values do affect the final output values. If there are j components and l features, j*l number of means and variances are to be initialized.

Step2: Initialize the parameter values for the common distribution covering all the data points in the given data set. Here calculate the mean and variance of each column in the data set. If there are l features, l number of means and variances are to be initialized.

Step3: Set the feature saliency of all the features to 0.5. If there are l features, l feature saliencies are to be initialized to the value 0.5.

Step4: Set the component weights of all the components equally such that sum is equal to one.

B. Units

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would

be the use of English units as identifiers in trade, such as "3.5-inch disk drive".

- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: "Wb/m2" or "webers per square meter", not "webers/m2". Spell out units when they appear in text: ". . . a few henries", not ". . . a few H".
- Use a zero before decimal points: "0.25", not ".25".

C. EXPECTATION STEP

In this step, various parameters corresponding to probability of each data point in relation with the component for a particular feature being relevant or irrelevant is calculated[1].

Before calculating, each parameter we check if any component is pruned in previous iterations and set all its corresponding parameters to zero instead of calculating the new values.

D. MAXIMIZATION STEP

Step5:

In this step, we calculate the parameters [1] which define each component (i.e mean and variance) and common distribution. We also calculate component weights and feature saliencies of the features.

Before calculating each parameter, we check if any component is pruned in previous iterations and set all its corresponding parameters to zero instead of calculating the new values.

E. PRUNING AND FEATURE RANKING

Step6:

In this step certain components get pruned. Certain parameters corresponding to mixtures and common distribution get pruned. The feature whose feature saliency have become 0 get the least rank among given ranks.

Step7: If component weight of any component becomes zero, prune it. It implies all parameters involving that component must be set to zero.

Step8: If feature saliency of any feature becomes zero, then mixture parameters involving that feature must be set to zero and that feature should be assigned the least rank in the available rank.

Step9: If feature saliency of any feature becomes one, then common distribution parameter involving that feature must be set to zero.

Repeat step 5 to 9 till certain iterations (we have considered 4 iterations)

F. RECORDING THE MODEL AND ITS MESSAGE LENGTH

The message length of the model is calculated and the parameters corresponding to the given model are stored.

Step10: calculate the message length

The first term in the above formula corresponds to log likelihood,

Second term corresponds to parameter code-length corresponding to $K\alpha_j$ values and D ρ_l values,

Third term corresponds to code length for calculating R parameters in each θ_{jl} where effective number of datapoints for estimating it is $N\alpha_j\rho_l$,

Similarly Fourth term is code length corresponding to parameters of common component.

Step11: Record the model along with its message length (here we have used 2-D array to store different models. Each row in the array corresponds to one model).

Step12: The component with lowest component weight is pruned.

Go to step5 till k (number of components) is less than k min (minimum number of components) which is given as input.

G. FINAL STEP

In this step feature ranks for the remaining features are set and the model with minimum message length returned.

Step13: Based on the feature saliency values of the last iteration the remaining features are assigned their ranks.

Step14: The minimum message length is found and respective model is returned.

IV. RESULTS AND DISCUSSION

UFSA has been compared against Relief-F evaluator[2] and Representation Entropy evaluator[3] using EM, Simple K-Means, Farthest First and Cobweb clustering techniques.

Five bench mark data sets Wine, Iris, Lenses, Bupa and Pima from the UCI Machine Repository[7] have been used for finding the effectiveness of UFSA algorithm. All the datasets are numerical datasets.

TABLE I. UCIML REPOSITORY BENCHMARK DATASETS

S.No	Data Set	Instances	Features	Number Of classes
1	Wine	178	13	3
2	Iris	150	4	3
3	Lenses	24	4	3
4	Bupa	345	6	2
5	Pima	414	8	3

UFSA algorithm is evaluated using EM, Simple k-Means, Farthest First and Cobweb clustering techniques on various feature subsets of a data set and the clustering error rate is used to measure the quality of the feature subset. For a data set of size 'n' the feature subset size may range from 1 to n. For instance ,the various feature subsets possible for Iris data set with size '4' are {4} , {4,3} , {4,3,1}, {4,3,1,2} for the standard ranking obtained by Relief-F and {3}, {3,4} , {3,4,1} , {3,4,1,2} for the ranking obtained by UFSA algorithm.

Graphs are plotted with error rates on the Y-axis and the number of significant features used for clustering on X-axis. The values on the X-axis can be interpreted as follows. When x is equal to 2 for a given data set, say Iris , it indicates that clustering is done with the two most important features 4th and 3rd of the Iris data set and the corresponding value on the Y-axis depicts the clustering error rate. The graph shows the error rates produced by clustering with the feature subsets obtained from the ranking of our algorithm as well as that obtained from Relief-F evaluator method. The performance of our algorithm, UFSA is close to and sometimes even better than that of Relief-F and Representation entropy.

A. Comparison with Relief-F evaluator

Table II shows the order of importance (feature ranking) obtained by UFSA and Relief-F evaluator for 5 datasets.

TABLE II. RANKING GIVEN BY RELIEF-F AND UFSA

Data Set	Relief-F	UFSA
Wine	12,7,13,1,10,6,11,2, 8,9,4,5,3	10,4,2,7,6,12,1,9,11,8, 3,13,5
Iris	4,3,1,2	3,4,1,2
Lenses	4,3,2,1	1,4,3,2
Bupa	3,5,6,4,1,2	5,3,4,6,2,1
Pima	5,2,4,3,7,1,8,6	2,6,4,1,8,7,3,5

For the wine dataset, in case of Farthest first clustering (fig 1), when the feature subset size is '1', '2' and '3' error rate is less when clustered using the feature ordering given by Relief-F than UFSA.

For feature subset of size '4','5', '6' UFSA performance is better compared to Relief-F. A lower error rate of 41.02% is obtained when we consider UFSA feature subset of size '5'. Among all the feature subsets, subset of size '5' has the least error rate . So we can consider the UFSA feature subset of size '5' for EM clustering.

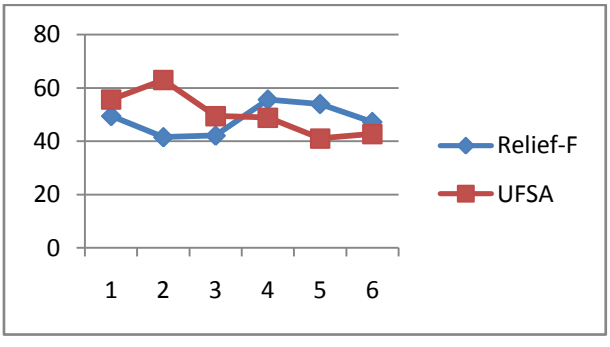

Figure 1. Comparison of UFSA and Relief-F for Wine dataset using Farthest First Clustering

In case of Cobweb Clustering (fig 2), the error rate is minimum (33.7%) for UFSA when compared with Relief-F (60.11%), when the most significant feature is considered. The error rates are varying when different sizes are used and is observed that error rates are minimum when subsets of size '1', '3', '5' and '6' are used and more for remaining subsets. Therefore the most significant feature can be considered for feature selection.

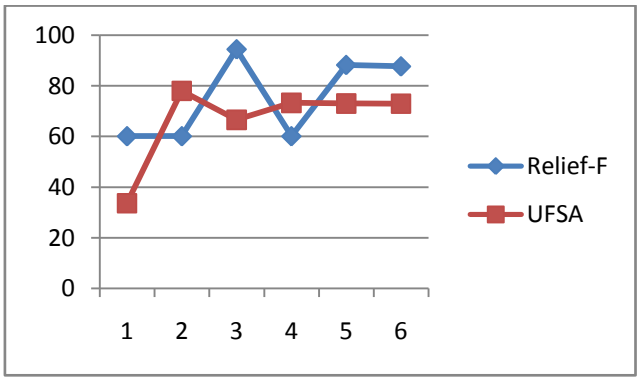

Figure 2. Comparison of UFSA and Relief-F for Wine dataset using Cobweb Clustering

For the Pima dataset, in case of k-means clustering (fig 3), when the feature subset size is '3' , error rate is slightly more for UFSA than Relief-F. For feature subset of size '1', '2', '4' and '5', UFSA performance is better compared to Relief-F. A lower error rate of 46.74% is obtained when we consider UFSA feature subset of size '2' , where error rate is 63.151% if we consider Relief-F. Therefore, among all the feature subsets, UFSA feature subset of size '2' can be considered for k-means clustering.

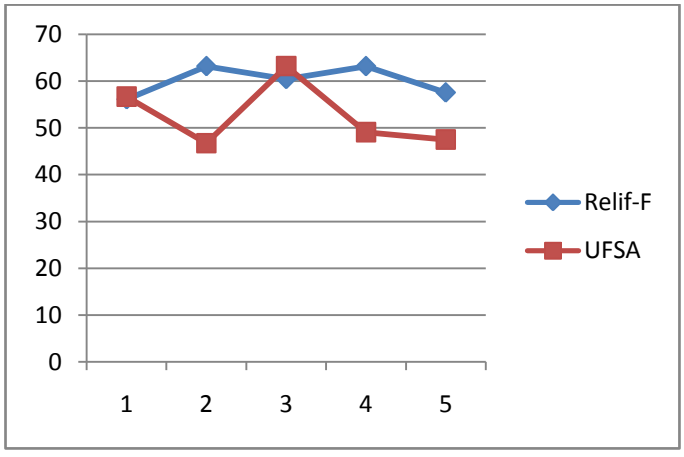

Figure 3. Comparison of UFSA and Relief-F for PIMA dataset using K-Means Clustering

In case of farthest first clustering (fig 4), when the feature subset size is '1', '2', '4' and '5', error rate is more for UFSA than Relief-F. For feature subset of size '3', UFSA performance is better compared to Relief-F. A lower error rate of 33.98% is obtained when we consider UFSA feature subset of size '1', '2', '3', '4'. Among all the feature subsets, UFSA feature subset of size '3' can be considered for farthest first clustering.

 ISSN: 2321-8169
150 – 155

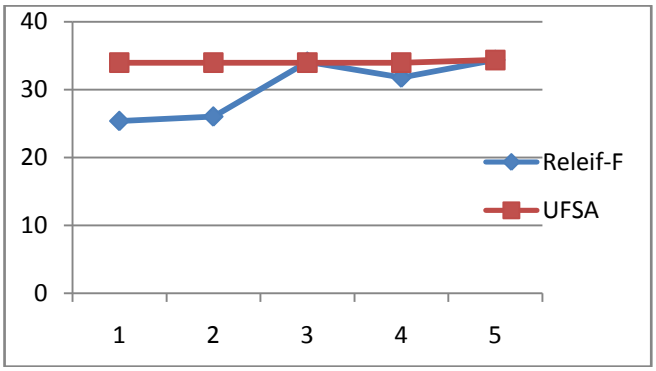

Figure 4. Comparison of UFSA and Relief-F for PIMA dataset using Farthest First Clustering

B. *Comparison with Representation Entropy Evaluator*

Table III shows the order of importance (feature ranking) obtained by UFSA and Representation entropy evaluator for the datasets.

TABLE III. RANKING GIVEN BY RELIEF-F AND UFSA

Data Set	Rep. Entropy	UFSA
Wine	13,5,12,11,9,8,6,3,1 ,7,2,10,4	10,4,2,7,6,12,1,9,11,8, 3,13,5
Iris	3,4,1,2	3,4,1,2
Lenses	1,4,3,2	1,4,3,2
Bupa	5,6,1,4,3,2	5,3,4,6,2,1

In case of Simple k-means clustering (fig 5), when the feature subset size is '1', '2' ,'3' and '6', error rate is more for UFSA than RepEnt. For feature subset of size '4' and '5', UFSA performance is better compared toRepEnt. A lower error rate of 21.9% is obtained when we consider UFSA feature subsets of size '4' and '5 among all other feature subsets. Therefore UFSA feature subset of size '4' can be considered for K-means Clustering.

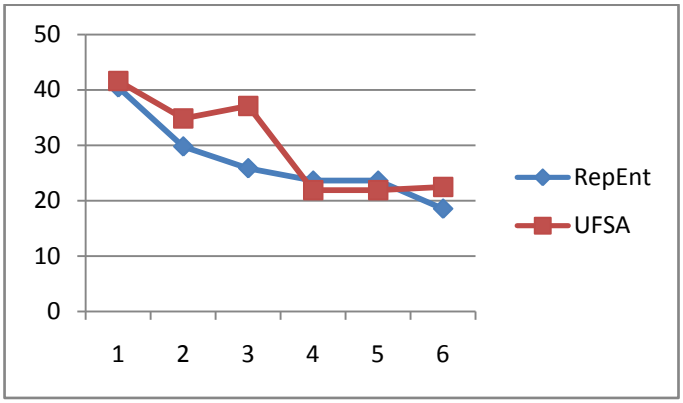

Figure 5. Comparison of UFSA and Rep Entropy for wine dataset using K-means Clustering

In case of Farthest First clustering (fig 6), when the feature subset size is '1', '2' ,'3' and '4', error rate is more for UFSA than RepEnt. For feature subset of size '5' and '6', UFSA performance is better compared to RepEnt. A lower error rate of 41.02% is obtained when we consider UFSA feature subset of size '5' among all other feature subsets. Therefore UFSA feature subset of size '5' can be considered for Farthest First Clustering.

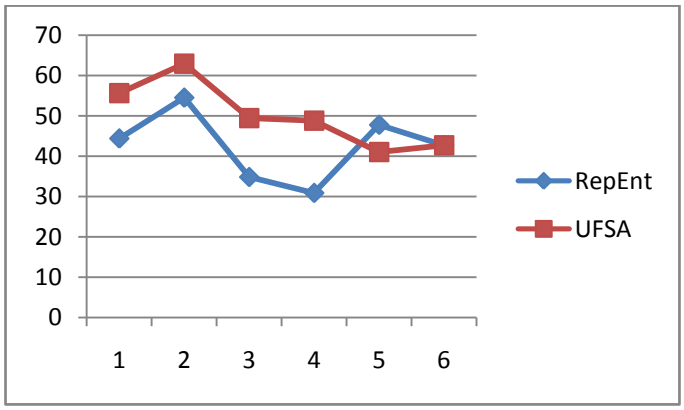

Figure 6. Comparison of UFSA and Rep Entropy for wine dataset using Farthest First Clustering

In case of Cobweb clustering (fig 7), when the feature subset size is '1', '2' ,'3', '4', '5' and '6' error rate is less for UFSA than RepEnt. Hence the overall performance of UFSA is better than RepEnt. A lower error rate of 33.7% is obtained when we consider UFSA feature subset of size '1' among all other feature subsets. Therefore UFSA feature subset of size '1' can be considered for Cobweb clustering.

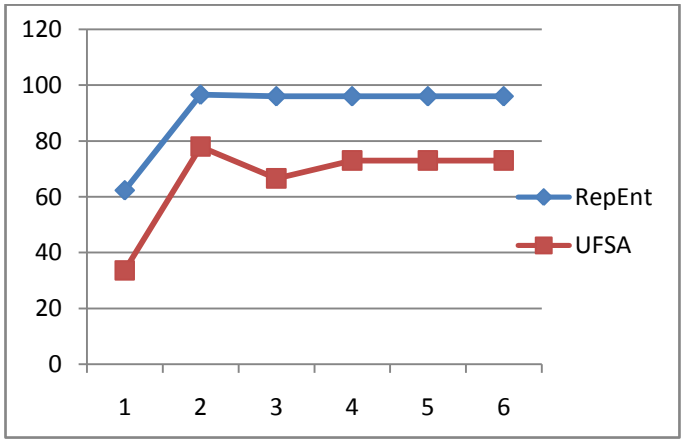

Figure 7. Comparison of UFSA and Rep Entropy for wine dataset using Cobweb Clustering

V. CONCLUSION AND FUTURE WORK

An algorithm for feature selection has been implemented which involves feature ranking based on unsupervised feature saliency approach in model based clustering. That is finding the order of importance of each feature which is used for discriminating clusters. This algorithm also returns one of the effective models among the various models generated by the algorithm.

The algorithm has been implemented in MATLAB[6]. The results (ranks) obtained from UFSA have been compared with the ranks obtained by Relief-F evaluator using four clustering techniques: EM, Simple K-Means, Farthest First and Cobweb. Also the results have been compared with the Representation Entropy algorithm which is an unsupervised technique to determine feature ranks. From the experimental study it is found that UFSA algorithm exceeds the performance of Relief-F and Representation Entropy algorithm in some cases.

The algorithm only works with the numerical datasets. This can be extended further to work with categorical and other datasets.

REFERENCES

[1] M.A.T. Figueiredo and A.K. Jain, "Unsupervised Learning of Finite Mixture Models," IEEE Trans. Pattern Analysis and MachineIntelligence, vol. 24, no. 3, pp. 381-396, Mar. 2002.
[2] K. Kira and L. Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," Proc. 10th Nat'l Conf. Artificial Intelligence (AAAI-92), pp. 129-134, 1992.
[3] V.Madhusudan Rao, and V.N.Sastry, "Unsupervised feature ranking based on representation entropy," Recent Advances in Information Technology,ISM Danbad pg no:514-518,March.2012.
[4] Jaiwen Han and Macheline Kamber, "Data Mining Concepts and Techniques," Morgan Kaufmann publishers,pages.383-434,2011
[5] M.H. Law, A.K. Jain, and M.A.T. Figueiredo, "Feature Selection in Mixture-Based Clustering," Advances in Neural Information Processing Systems 15, pp. 625-632, Cambridge, Mass.: MIT Press, 2003.
[6] http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf
[7] http://archive.ics.uci.edu/ml/datasets.html