Transaction on Evolutionary algorithm and Clustering
ISSN: 2229-8711 Online Publication, June 2011
www.pcoglobal.com/gjto.htm
CA-O34/GJTO

Global Journal of Technology & Optimization
Volume 2, 2011

ON INITIALIZATION OF THE EXPECTATION-MAXIMIZATION CLUSTERING ALGORITHM

Z. Volkovich, R. Avros, and M. Golani

Software Engineering Department, ORT Braude College of Engineering, Karmie, Israel
Email: vlvolkov@braude.ac.il, r_avros@braude.ac.il, matig@braude.ac.il

Received January 2011, Revised March 2011, Accepted April 2011

Abstract
Iterative clustering algorithms commonly do not lead to optimal cluster solutions. Partitions that are generated by these algorithms are known to be sensitive to the initial partitions that are fed as an input parameter. A "good" selection of initial partitions is an essential clustering problem. In this paper we introduce a new method for constructing the initial partitions set to be used by the Expectation-Maximization clustering algorithm (*EM* algorithm). Our approach follows ideas from the Cross-Entropy method. We use a sample clustering provided by means of the *EM* algorithm as an alternative for the simulation phase of the Cross-Entropy method. Experimental results reflect a good performance with respect to the offered method.

Keywords: *EM*-algorithm, Iterative clustering algorithms, Clustering initialization

1. Introduction

Clustering methods are widely used in many different areas as a practical tool to understand hidden structures in complex data. The clustering goal is primarily to group jointly similar items. Consequently, it is assumed that beside the observed variables of each data item, there is a hidden, unseen variable representing the "cluster membership" of that item. A variety of iterative clustering procedures (e.g., $k-$means, and the *EM* algorithm) requires an initial partition of a dataset as an input parameter. In iterative clustering approaches, the quality of the generated partitions is strongly dependent on this initial partitions choice. An efficient selection of an initial partition is therefore, a must have requirement – since the paramount importance for successful implementation of clustering algorithms. This problem has been considered in many works (see, e.g. [1-3]).

This paper proposes new method for constructing initial partitions to be used by the Expectation-Maximization clustering algorithm (*EM* algorithm). The presented approach follows ideas from the Cross-Entropy method, where we use a sample clustering produced by means of the *EM* algorithm as an alternative for the simulation phase. The provided Experimental results reflect a good performance of the offered method.

The paper is organized as follows. The Gaussian Mixture Model framework is reviewed in Section2. Section3 provides a short description of the refinement clustering algorithm. Section4 summarizes the Cross-Entropy method, and discusses its advantages and drawbacks. It also introduces the new initialization procedure. Numerical experiments are presented in Section5.

2. The Gaussian Mixture Model

Many clustering methods are based on a density estimation perception. Thus, the data is considered to be independently extracted from a mixed population while the mixing labels (cluster identifiers) are hidden. More specifically, if we consider the data $\{x_1,...,x_m\}$ to be a set of vectors in a subset *X* of the*n*-dimensional Euclidean space R^n having clusters $\{C_j\}, j=1,...,k$, then the underlying distribution μ of *X* is assumed to be written as

$$\mu = \sum_{j=1}^{k} p_j \mu_j,$$

where set $P=\{p_j, j=1,...,k\}$ is the cluster probabilities and $\mu_j, j=1,...,k$ are the inner clusters distributions. The wide spread *EM* clustering algorithm suggests the Gaussian Mixture Model (*GMM*) of data fitting [5]. In this case, the distributions $\mu_j, j=1,...,k$ are identified by multivariate Gaussian distributions $G(x \mid y_j, \sigma_j)$ where $Y=\{y_j, j=1,...,k\}$ and covariance matrices $\Sigma=\{\sigma_j, j=1,...,k\}$. Thus, the clusters are ellipsoidal sets which are centered at the means y_j, such that the covariance matrices σ_j provide the clusters' shape. Usually, these parameters are estimated from the data, and can be either allowed to vary between clusters, or guarded to be the same for all clusters. A categorization of several covariance models can be found in [6], and was presented also in [5].

In the presence of incomplete data, the *EM* algorithm [5] is an iterative procedure that maximizes the log likelihood function. Here, the 'complete' data is considered to be $s_i=(x_i,\mathbf{z}_i),\ i=1,...,m,\ \mathbf{z}_i=(z_{i1},...,z_{ik})$, where for *i*=1,..,*m*, *j*=1,…,*k*

$$z_{ij}=\begin{cases}1 \text{ if } x_i \text{ belongs to group } j,\\ 0 \qquad\qquad \text{otherwise.}\end{cases}$$

The resulting complete-data log likelihood is:

$$l(\mathbf{P},\mathbf{Y},\Sigma)=\sum_{i=1}^{m} z_{ij}\log\left(\sum_{j=1}^{k} p_j G\left(x_i \mid y_j,\sigma_j\right)\right). \quad (1)$$

The algorithm starts from a random initialization of the hidden variables $\mathbf{z}_i$ and iterates between the *E*-step, where these variables are evaluated from the data with the present parameter values, and the *M*-step in which Equation 1 is maximized with respect to the parameters. The standard k *-means* algorithm can be viewed as a partial version of the *EM* algorithm corresponding to the uniform spherical Gaussian model, with equal sized clusters. The well recognized drawback of the *EM* algorithm is that it fails to converge to the global maximum. A simple and a standard way for handling this drawback, is by multiple runs of the algorithm, with different initial partitions. Often, a large number of re–runs is required, making the algorithm time complexity relatively high.

3. The Refine Algorithm

One of most successful initialization approaches for iterative refinement clustering algorithms was provided by U. Fayyad, C. Reina, and P.S. Bradley [4, 7]. The procedure is based on an efficient technique for estimating the distribution modes, and can be applied to various iterative clustering algorithms. The applicability of this method to the *EM* algorithm was demonstrated by the authors, and it was shown that refined initial centroids do indeed lead to improved partitions. This "Refine" algorithm can be described as follows:

Input arguments:

- $k-$ the number of clusters;
- $SP-$ a vector of k initial centroids;
- $Data-$ the dataset;
- $J-$ the number of samples;
- $M-$ the sample size.

The algorithm draws J small samples $S_j, j=1,...,J$, and applies the *EM* algorithm to generate a partition having *k* centroids CM_j. In case that some of the centroids are identical (i.e. the corresponding partition contains less than *k* clusters), then the sample S_j is clustered again with a different initial vector *SP*. The union of centroids $CM_j, j=1,...,J$ is denoted as *CM*. The set *CM*is clustered by the*k*-means algorithm with the initial centroids CM_j, the centroids of the resulting partition are denoted by $FM_j, j=1,...,J$ and

$$FMS=\bigcup_{j=1}^{J} FM_j.$$

The final set of *k* centroids *FM* is the set FM_j maximizing the likelihood with respect to *CM*. A pseudo-code version of the algorithm is as follows:

---

1. *CM*=∅;
2. for $j=1,...,J$
   (a) Draw sample S_j having size *M* from *Data*;
   (b) $CM_j=EM(SP,S_j,k)$;
   (c) $CM=CM\cup CM_j$.
3. *FMS*=∅
4. for $j=1,...,J$
   (a) $FM_j=KMeans(CM_j,CM,k)$
   (b) $FMS=FMS\cup FM_j$
5. $FM=Arg\max\{Likelihood(FM_j,CM)\}$
6. $Return(FM)$

---

4. Sequential Initialization of the *EM* Algorithm

As mentioned above, usage of the *EM* algorithm for clustering can be considered as a method for solving an optimization problem which consists of the maximization of the log likelihood function (Equation (1)). One of the generic methods for this purpose is the Cross-Entropy (*CE*) method. *CE* finds many applications in different research fields (see the *CE* site *http://iew3.technion.ac.il/CE/about.php*). Generally speaking, the essence of the *CE* method is the following:

1. Generate a sample of random data that fits parameters of the underlying distribution.
2. Update the parameters in order to produce a "better" sample in the next iteration.
3. Iterate the procedure until the process is "stabilized".

Application of *CE* to clustering and vector quantization has been provided in [8]. The method has been shown to be robust with respect to the choice of initial centroids. This task has been considered as an optimization problem

$$L=\min_{c_1,...,c_k} R(c_1,...,c_k)=\min_{c_1,...,c_k}\sum_{j=1}^{k}\sum_{x\in C_j}\left\|x-c_j\right\|^2, \quad (2)$$

where $c_1,...,c_k$ are the decision variables, in this case - centroids of the clusters $\{C_j\}, j=1,...,k$.

In clustering applications, Step 1 of the procedure is performed by simulation [8]. However, the *CE* method has several disadvantages that are related mainly to the simulation phase. Specifically, a simulation task appears to be computationally expensive for high dimensional data. All simulations are performed under the assumption that the underlying distribution can be properly approximated by means of the Gaussian distribution. This assumption is rarely satisfied for real data, which is often sparse; as it appears in text mining applications [9]. Evidently, modeling of such a high-dimensional dataset, by means of a mixed normal law, can lead to a large deviation from the underlying distribution. This is an aspect of the so-called "curse of dimensionality".

The method proposed here avoids the simulation step presented in the *CE* approach. Instead, we consider the clustering task as an optimization problem with the objective function from Equation (1). Consequently, in a similar manner to the above described "Refine" Algorithm (see Section 3), we use the log likelihood value which is calculated according to Equation (1) as a criterion for "elite" samples selection instead of the concentration measures of [8] by means of Equation (2). Actually, we consider a more general optimization problem.

As in the *CE* approach, we start by outlining the parameters of our algorithm:

- $k-$ the number of clusters;
- $N-$ the number of drawn samples;

- M – the sample size;
- ρ – the fraction of correct ("elite") samples.

The algorithm consists of the following steps:

1. Draw *N* random samples $S_i,\ i=1,...N$, of size *M* from the dataset, and set counter *co*=1.
2. Apply the *EM* algorithm to the samples, and generate clusters with centroids y_{ji}, and covariance matrices $\Sigma_{ji},\ j=1,...,k,\ i=1,...N$ for the partitions $\Pi_i(S_i),\ i=1,...,N.$ Initial parameters of the *EM* algorithm are chosen randomly for $co=1$, and are borrowed from iteration *co*-1 if*co*>1. Recall, that random initialization is generated by a random assignment of the dataset elements to clusters. The *GMM* parameters (see Section 2) are calculated according to this assignment.
3. Calculate the partition quality values $L_i,\ i=1,...,N$, according to Equation (1).
4. Rank the sequence $L_i,\ i=1,...,N$ and take $N^{elite}=[\rho N]$ "elite" samples corresponding to the biggest $[\rho N]$ values of L_i, say, $S_1,...,S_{N^{elite}}$.
5. Re-compute the *GMM* model parameters involved in the *EM* algorithm by means of the algorithm for partition of the united sample
$$\hat{S}_{co}=\bigcup_{i=1}^{N^{elite}} S_i$$
(The choice of *EM* initial parameters is indicated in Step 2 above).
6. If the stop criterion is met, then stop and accept the obtained *GMM* parameters as an estimate for the true *GMM* parameters. Otherwise, set *co* = *co*+1 and go to Step 2.

A stopping criterion can be formulated in terms of process stabilization. As a first decisive factor, we can consider -in terms of stabilization - the log likelihood empirical values. Specifically, it is possible to calculate this value for $\widehat{L}_{co}$ according to the *EM*- partitions obtained for the sets $\widehat{S}_{co}$ in Step5. The process is stopped if the difference between two consecutive values $\widehat{L}_{co}$ and $\widehat{L}_{co+1}$ is within some predefined tolerance.

Another criterion deals with the solution stabilization. Note that such a rule is going to be more complicated in comparison with *CE* clustering. The main challenge is the inherent symmetry of clusters with respect to their labels permutation, that leads to cluster correspondence problems in the two samples $\widehat{S}_{co}$ and $\widehat{S}_{co+1}$.

Here, in order to compare two solutions, we assign each item of the dataset *X* to the clusters of the *EM*- partitions of $\widehat{S}_{co}$ and $\widehat{S}_{co+1}$ according to the maximal probabilities calculated by means of the *GMM* with the obtained parameters. Let us denote these assigns by α_{co} and α_{co+1}. An element of the dataset may be labeled differently. Accordingly, the clusters' labeling of the mentioned *EM* solutions can be permuted in a different way. We solve the labeling corresponding problem by resting upon a natural suggestion that the most intersected clusters induced in the dataset by the partitions of $\widehat{S}_{co}$ and $\widehat{S}_{co+1}$ correspond to one another. So, we look over all possible cluster labels permutations for a permutation that minimizes the actual misclassification between two sequential steps.

Specifically, let us denote Ψ_k as the collection of all possible permutations for the set $\{1,2,...,k\}$. A favored permutation ψ^*_{co} has to provide the smallest misclassification between the two classifications, i.e.
$$\psi^*_{co}=\operatorname{argmin}(\psi\in\Psi_k \mid D(\alpha_{co}(X),\psi(\alpha_{co+1}(X))),$$
where*D* is the misclassification measure:
$$D(\alpha_{co}(X),\psi(\alpha_{co+1}(X)))=\frac{1}{|X|}\sum_{x\in X}\chi(\alpha_{co}(x),\psi(\alpha_{co+1}(x)))$$

Here, $\chi(\alpha_{co}(x),\psi(\alpha_{co+1}(x)))$ is an indicator function of the event $\alpha_{co}(x)\neq\psi(\alpha_{co+1}(x))$. The straightforward solution for this problem requires testing all $k!$ possible permutations. On the other hand, this task is a special case of the minimum weighed perfect bi-variant matching problem, which can be solved by the well-known Hungarian method with computational complexity $O(k^3)$ [10]. Furthermore, to express a stopping criterion of the algorithm, we compare the sets $\Theta_{co}=\{(y_i^{(co)},\sigma_i^{(co)}),\ i=1,...,k\}$ of the *GMM* parameters, that were found by the algorithm in the step *co*. actually, the convergence of Θ_{co} for $co\to\infty$ is a desired criterion and we can stop the process if the value
$$\Delta_{co}=w_1\sum_{i=1}^{k}(y_i^{(co)}-y_{\psi^*_{co}(i)}^{(co+1)})^2+w_2\sum_{i=1}^{k}\|\sigma_i^{(co)}-\sigma_{\psi^*_{co}(i)}^{(co+1)}\|^2$$ is sufficiently small. Here, $\|\bullet\|^2$ stands for the L^2 matrixes norm, and w_1,w_2 are the control coefficients. A different criterion aiming to bound L_∞ norm of the diagonal elements of the obtained covariance matrices is also available in the literature [8].

5. Experimental Results

Evaluation of the described methodology has been provided by numerical experiments on real datasets. First, we consider the well known Iris Flower Dataset available at
$http://fmwww.bc.edu/ec-p/data/micro/iris.dta$.
This dataset contains features of three different classes of flowers:

- 0 - Iris Setosa,
- 1 - Iris Versicolour,
- 2 - Iris Virginica.

There are 50 examples for each class in the dataset. Each example has a four dimensional feature vector. The obtained clustering results are evaluated by the well known Rand index [11]. In addition,the partition quality was evaluated by means of the "misclassified" items number which we denote by *Dif*. We compare four initial partition building techniques. Those are:

- *EM*+ - clustering via *EM* algorithm using our sequential

initialization procedure (see Section 4), with assignment to the nearest centroid;

- *Refine* - clustering via *EM*- algorithm using the initialization by the Algorithm Refine of U. Fayyad, C. Reina, and P.S. Bradley in (see Section 3),
- *EMM*- clustering via *EM* algorithm, using our sequential initialization procedure, with final clustering of the data by means the standard *EM* algorithm,
- *EM*- a data clustering by means the standard *EM* algorithm, without any additional initialization procedures.

The outcomes shown in the following tables demonstrate good performance of clustering via EM_+ - algorithm based on our sequential initialization procedure.

Table 1: Data set Iris, $K = 3$, $R = 40$, $\rho = 0.2$

$N = 40$	Rand	Dif	Dif (%)	$N = 60$	Rand	Dif	Dif (%)
EM+	0.982	14	9.33	EM+	0.971	3	2
REFINE	0.957	5	3.33	REFINE	0.953	5	3.33
EMM	0.957	5	3.33	EMM	0.953	5	3.33
EM	0.785	61	40.67	EM	0.507	63	42

The second dataset is selected from the text collections available at

http : / / *www.dcs.gla.ac.uk* / *idom* / *ir_resources* / *test_collections* / .

It consists of the following three text collections:

- DC0–Medlars Collection (1033 medical abstracts),
- DC1–CISI Collection (1460 information science abstracts),
- DC2–Cran field Collection (1400 aerodynamics abstracts).

This dataset has been considered in a number of papers (e.g. [12]). Following the well known "bag of words" vector space model (e.g. [9]) 300 and 500 "best" terms were selected (see [13] for term selection details). This dataset is known to be well separated with the help of the two leading principal components. This representation was used in our experiments.

Table 2: Three text collection represented by300 terms, $K = 3$, $R = 30$, $\rho = 0.25$

$N = 30$	Rand	Dif	Dif (%)	$N = 40$	Rand	Dif	Dif (%)
EM+	0.962	114	2.93	EM+	0.951	150	3.86
REFINE	0.935	206	5.29	REFINE	0.935	206	5.29
EMM	0.935	206	5.29	EMM	0.935	206	5.29
EM	0.745	1380	35.47	EM	0.935	206	5.29

Table 3: Three text collection represented by500 terms, $K = 3$, $R = 60$, $\rho = 0.25$

$N = 40$	Rand	Dif	Dif (%)	$N = 60$	Rand	Dif	Dif (%)
EM+	0.968	95	2.44	EM+	0.973	82	2.11
REFINE	0.962	118	3.03	REFINE	0.962	118	3.03
EMM	0.753	1358	34.90	EMM	0.962	118	3.03
EM	0.962	117	3.01	EM	0.962	118	3.03

References

[1] A. Juan and E. Vidal, 2000. Comparison of Four Initialization Techniques for the K -Medians Clustering Algorithm, Advances in Pattern Recognition, Lecture Notes in Computer Science, 1876, pp. 842-852.

[2] R. Messina and D. Jouvet, 2004.Sequential clustering algorithm for Gaussian mixture initialization, Acoustics, Speech and Signal Processing, In Proceedings of the IEEE International Conference (ICASSP'04), pp. 833-836.

[3] J. Sheu, W.-M, Chen, W.-B, Tsai and K.-T- Chu, 2010.An intelligent initialization method for the k-means clustering algorithm, International Journal of Innovative Computing, Information and Control (ICIC International) ISSN 1349-4198, 6(6), pp. 2551–2566

[4] U.M. Fayyad, C. Reina, and P.S. Bradley, 1998, Initialization of iterative refinement clustering algorithms, In Knowledge Discovery and Data Mining, pp. 194-198.

[5] C. Fraley and A.E. Raftery, 1998, How many clusters? Which clustering method? Answers via model-based cluster analysis, The Computer Journal, 41(8), pp. 578-588

[6] J.D. Banfeld and A.E. Raftery, 1993.Model-based Gaussian and non-Gaussian clustering, Biometrics, 49, pp.803-821

[7] P.S. Bradley and U.M. Fayyad, 1998. Refining initial points for K-Means clustering, In Proc. 15th International Conf. on Machine Learning, pp. 91-99.Morgan Kaufmann, San Francisco, CA

[8] D. Kroese, R. Rubinstein and T. Taimre, 2006. Application of the cross-entropy method to clustering and vector quantization .Journal of Global Optimization 37, pp.137-157

[9] M. Berry and M. Browne, 1999.Understanding Search Engines, SIAM

[10] H. Kuhn, 1955. Hungarian method for the assignment problem, Naval Research Logistics Quarterly, 2, pp. 83-97

[11] W. Rand, 1971. Objective criteria for the evaluation of clustering methods, Journal Am. Stat. Assoc., 66, pp. 846-850.

[12] V. Volkovich, J. Kogan, and C. Nicholas, 2004, $k-$means initialization by sampling large datasets. In I. Dhillon and J. Kogan, editors, Proceedings of the Workshop on Clustering High Dimensional Data and its Applications (held in conjunction with SDM 2004), pp. 17-22.

[13] I. Dhillon, J. Kogan, and C. Nicholas, 2003. Feature selection and document clustering. In M. Berry, editor, A Comprehensive Survey of Text Mining, pp. 73-100. Springer, Berlin