Proceedings of the 2004 Winter Simulation Conference
R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

GLOBAL LIKELIHOOD OPTIMIZATION VIA THE CROSS-ENTROPY METHOD WITH AN APPLICATION TO MIXTURE MODELS

Zdravko Botev
Dirk P. Kroese

Department of Mathematics
The University of Queensland
Brisbane 4072, Australia

ABSTRACT

Global likelihood maximization is an important aspect of many statistical analyses. Often the likelihood function is highly multi-extremal. This presents a significant challenge to standard search procedures, which often settle too quickly into an inferior local maximum. We present a new approach based on the cross-entropy (CE) method, and illustrate its use for the analysis of mixture models.

1 INTRODUCTION

Many statistical problems involve the maximization of the *likelihood function*, which, for each choice of model parameters gives the probability (or density) of the observed data. A typical example occurs in *cluster analysis* where the data are assumed to come from a *mixture* of (usually) Gaussian densities; and the objective is to estimate the parameters of this mixture by maximizing the likelihood function. Direct optimization of the likelihood function in this case is not a simple task, due to the constraints on the parameters, and, more importantly, the complicated nature of the likelihood function, which in general has a great number of local maxima and saddle-points.

Traditional local search methods such as the gradient-based quasi-Newton method are often inadequate, because they usually fail to find the global maximum of the likelihood function, and in some cases fail to converge altogether. The classic Nelder-Mead method (Nelder and Mead 1965), which evaluates the function at the vertices of a simplex and then iteratively shrinks the simplex as better points are found until some desired bound is obtained, may have the same problems. Moreover, this method is formulated for unconstrained optimization problems only.

A popular method to estimate the parameters of the mixture model is the well-known *EM algorithm*; see for example (Dempster, Laird, and Rubin 1977) and (McLachlan and Krishnan 1997). The EM algorithm is very fast and general, but can only be guaranteed to converge to a local maximum, under certain continuity conditions; see for example (Wu 1983) and (Boyles 1983). Moreover, the convergence of the algorithm depends strongly on the starting values. The correct choice of starting values may not always be clear. This difficulty is shared by many "global" search algorithms, such as the *genetic algorithm* (Goldberg 1989), where starting values are often picked "at random".

In this paper we present a new approach to likelihood maximization which is based on the well-known *cross-entropy* (CE) method (Rubinstein and Kroese 2004). The purpose of this paper is to

1. explain how the CE method can be employed as a global likelihood optimization procedure for mixture models in cluster analysis,
2. introduce a useful modification of CE called the *injection method*,
3. compare the CE approach with the classical EM approach, for mixture models in cluster analysis.

The CE method has been successfully applied to a great variety of discrete, i.e., combinatorial, optimization problems, with both deterministic and random (noisy) objective functions; see for example (Alon, Kroese, Raviv, and Rubinstein 2004), (de Boer, Kroese, Mannor, and Rubinstein 2004), (Chepuri and de Mello 2004), (Dubin 2002), (Helvik and Wittner 2001), (Liu, Doucet, and Singh 2004), (Mannor, Rubinstein, and Gat 2003), (Margolin 2002), (Rubinstein 1999), (Rubinstein 2002), (Rubinstein 2001), (O. Wittner and B. E. Helvik 2002). As a consequence, the behavior of the CE method is fairly well understood, at least from a pragmatic point of view; see also (Margolin 2004).

However, for continuous optimization problems – likelihood maximization being a typical example – much less is known about the behavior of the algorithm, although the main ideas are described in (Rubinstein and Kroese 2004).

Standard implementations of the CE method sometimes have the problem that the sampling distributions "shrink" too fast to a degenerate (atomic) distribution, preventing the method from finding the good solution. We tackle this problem by "injecting" extra noise into the sampling distributions, at certain stages of the algorithm. This significantly improves the performance in terms of accuracy.

We illustrate numerically the performance of the CE method with that of the classic EM method, for a typical cluster problem. We find that, when the likelihood function is ill-behaved and highly constrained, the CE method finds superior solutions to those found by the EM method. This is typically the case when the number of points is not too large and the clusters are overlapping. Moreover, we find that the CE algorithm is quite robust under different starting conditions, whereas the EM may require many random initial guesses before it converges. The main advantage of EM over CE is its fast convergence.

The outline for the rest of the paper is as follows. In Section 2 we start with the basic setting of clustering analysis via mixture models, and describe how the EM algorithm can be employed to estimate the model parameters. The main CE algorithm is given in Section 3 along with various modifications, including variance injection. In Section 4 we compare the CE and EM algorithms for a 2-dimensional clustering problem. Finally, in Section 5 we list our conclusions and possible directions for future research.

2 CLUSTERING VIA EM

We recall the basic setting of clustering problems. The data consists of a collection of points $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ in some d-dimensional Euclidean space. We assume that the data in $\mathcal{Y}$ are the outcomes of i.i.d. random vectors[1] $\mathbf{Y}_1, \ldots, \mathbf{Y}_n$, each having a *mixture* density

$$f(\mathbf{y}; \boldsymbol{\theta}) = \sum_{c=1}^{k} w_c\, f_c(\mathbf{y}; \boldsymbol{\eta}_c), \qquad (1)$$

where $\boldsymbol{\theta} = (\mathbf{w}, \boldsymbol{\eta})$ is an unknown parameter vector, which includes the weights $\mathbf{w} = (w_1, \ldots, w_k)$ and the vector $\boldsymbol{\eta} = (\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_k)$ containing all the parameters of the densities $\{f_c(\cdot; \boldsymbol{\eta}_c)\}$. The standard example is where each density f_c is Gaussian with unknown expectation vector $\boldsymbol{\mu}_c$ and covariance matrix Σ_c. A fundamental approach to estimating the parameter $\boldsymbol{\theta}$ from the data $\mathcal{Y}$ is to choose the estimate such that the likelihood function

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{Y}) := \prod_{i=1}^{n} f(\boldsymbol{y}_i; \boldsymbol{\theta}) \qquad (2)$$

(or, equivalently, its logarithm) is maximized. However, finding this *maximum likelihood* estimate is in general not easy for these mixture models, since the likelihood function $\mathcal{L}$ is typically multi-extremal.

A different approach is to estimate $\boldsymbol{\theta}$ using the well-known *EM method* (McLachlan and Krishnan 1997). Here one views the data $\mathcal{Y}$ as only the *observed* part of a more complete data set $\{\mathcal{C}, \mathcal{Y}\}$. Namely, we may generate each random vector $\mathbf{Y}$ via a two-step procedure: first draw a random variable $C \in \{1, \ldots, k\}$ according to probabilities $\{w_1, \ldots, w_k\}$ and then, given $C = c$, draw $\mathbf{Y}$ from f_c. Using this point of view, we can interpret the data $\mathcal{Y}$ as only a part of the true data $\{\mathcal{C}, \mathcal{Y}\}$, where $\mathcal{C} = \{C_1, \ldots, C_n\}$. The value of C_i – which indicates from which distribution $\mathbf{Y}_i$ was drawn – remains *hidden.*

If $\boldsymbol{\theta} = \boldsymbol{\theta}^o$ were *known*, then assessing to which density f_c each point $\mathbf{y}$ belongs would be easy. Namely, the conditional distribution of C (dropping the index) given $\mathbf{Y} = \mathbf{y}$ is by Bayes's formula equal to

$$p^o(c\,|\,\mathbf{y}) := \mathbb{P}_{\boldsymbol{\theta}^o}(C = c\,|\,\mathbf{Y} = \mathbf{y}) = \frac{w_c^o\, f_c(\mathbf{y}; \boldsymbol{\eta}_c^o)}{f(\mathbf{y}; \boldsymbol{\theta}^o)}, \qquad (3)$$

for $c = 1, \ldots, k$. For a given *guess* $\boldsymbol{\theta} = \boldsymbol{\theta}^o$ one could compute instead of the logarithm of the likelihood function (1), the *expected* log-likelihood function

$$\mathbb{E} \log \mathcal{L}(\boldsymbol{\theta}; \mathcal{C}, \mathcal{Y}) := \mathbb{E} \sum_{i=1}^{n} \log(w_{C_i}\, f_{C_i}(\mathbf{y}_i; \boldsymbol{\eta}_{C_i})),$$

where the $\{C_i\}$ are independent and distributed according to $\{p^o(c|\mathbf{y}_i)\}$ in (3). This is the so-called *E-step* of the EM algorithm. In the *M-step* we maximize the expected log-likelihood with respect to the w_c and $\boldsymbol{\eta}_c$. That is, we maximize

$$\mathbb{E} \log \mathcal{L}(\boldsymbol{\theta}; \mathcal{C}, \mathcal{Y}) = \sum_{i=1}^{n} \mathbb{E}\left[\log w_{C_i} + \log f_{C_i}(\mathbf{y}_i; \boldsymbol{\eta}_{C_i})\right]$$

$$= \sum_{i=1}^{n} \sum_{c=1}^{k} p^o(c\,|\,\mathbf{y}_i) \left[\log w_c + \log f_c(\mathbf{y}_i; \boldsymbol{\eta}_c)\right] , \qquad (4)$$

under the condition that $\sum_c w_c = 1$. Using Lagrange multipliers in (4) and the fact that $\sum_c p^o(c\,|\,\mathbf{y}_i) = 1$, gives the maximum likelihood estimate (MLE)

$$\hat{w}_c = \frac{1}{n} \sum_{i=1}^{n} p^o(c\,|\,\mathbf{y}_i) . \qquad (5)$$

Finding the MLE for $\boldsymbol{\eta}_c$ follows now from optimizing $\sum_i p^o(c\,|\,\mathbf{y}_i) \log f_c(\mathbf{y}_i; \boldsymbol{\eta}_c)$. For the Gaussian case this leads to the formulas

$$\hat{\boldsymbol{\mu}}_c = \frac{\sum_{i=1}^{n} p^o(c\,|\,\mathbf{y}_i)\, \mathbf{y}_i}{\sum_{i=1}^{n} p^o(c\,|\,\mathbf{y}_i)}, \qquad (6)$$

[1] We will always interpret vectors as *column* vectors.

and

$$\hat{\Sigma}_c = \frac{\sum_{i=1}^{n} p^o(c\,|\,\mathbf{y}_i)\,(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_c)(\mathbf{y}_i - \hat{\boldsymbol{\mu}}_c)^{\mathrm{T}}}{\sum_{i=1}^{n} p^o(c\,|\,\mathbf{y}_i)}, \qquad (7)$$

which are very similar to the well-known formulas for the MLEs of the parameters of a Gaussian distribution. The EM algorithm in the Gaussian case now consists of iterating equations (3), (5), (6) and (7) until convergence is reached.

Note that the EM algorithm is a local search procedure and therefore there is no guarantee that it converges to the global maximum. Indeed, in some cases the global maximum may be infinity; see below.

3 CLUSTERING VIA CE

As an alternative to the EM algorithm we consider the CE approach, where we view the clustering problem as a continuous multi-extremal optimization problem with constraints. Specifically, we wish to maximize the log-likelihood function

$$\log \mathcal{L}(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log f(\boldsymbol{y}_i;\boldsymbol{\theta}) \qquad (8)$$

over the set Ω of all possible $\boldsymbol{\theta}$. At this point it is important to mention that if Ω is chosen as large as possible – i.e., any mixture distribution is possible – then the global maximization of (8) is an ill-posed problem! Namely, by choosing "point" or "line" clusters, or in general "degenerate" clusters, one can make the value of the (log-)likelihood infinite. It is therefore useful to restrict the parameter set in such a way that degenerate clusters (sometimes called spurious clusters) are not allowed.

For an introductory treatment of the concepts and theory behind the CE method we refer to the CE tutorial (de Boer, Kroese, Mannor, and Rubinstein 2004), which is also available on-line from the CE homepage at

`http://www.cemethod.org`

In this paper we will only explain the relevant ideas with respect to optimizing (8), which are quite intuitive.

Consider for simplicity the clustering problem with dimension $d = 2$. We may assume that $\boldsymbol{\theta}$ is vector in $(6k-1)$-dimensional space. Namely, apart from a total of $k-1$ weights (one weight can be omitted since the sum of the weights is 1), each of the k clusters is associated with 2 means, 2 standard deviations and 1 correlation coefficient. Let us assume that $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_{6k-1})^{\mathrm{T}}$ is such that $\theta_1,\ldots,\theta_{2k}$ are associated with the means, $\theta_{2k+1},\ldots,\theta_{4k}$ with the standard deviations, $\theta_{4k+1},\ldots,\theta_{5k}$ with the correlation coefficients and the remaining θs with the weights. Then, we have a constrained optimization problem over the convex set $\Omega \subset \mathbb{R}^{6k-1}$ with

$$\begin{array}{llll} \theta_i^{\text{low}} & \le & \theta_i, & i = 2k+1,\ldots,4k \\ -\rho_i^{\text{low}} & \le & \theta_i \le \rho_i^{\text{high}}, & i = 4k+1,\ldots,5k, \\ 0 & \le & \theta_i \le 1, & i = 5k+1,\ldots,6k-1\ . \end{array}$$

Here, the θ_i^{low} and θ_i^{high} specify the lower- and upper-bounds for the variances and correlation coefficients; this in view of the "degeneracy" problem discussed before.

The basic procedure of the CE method is to iteratively

(a) generate random samples in Ω according to a specified sampling distribution, followed by

(b) updating of these parameters on the basis of the best scoring samples, in order to produce better scoring samples in the next iteration.

The updating rules follow from cross-entropy minimization and often have a simple form.

In this paper we take the sampling distribution to be (truncated) Gaussian with independent components. That is, with each parameter θ_i in $\boldsymbol{\theta}$ we associate a 1-dimensional Gaussian distribution $\mathsf{N}(a_i, b_i^2)$. The updating rule in (b) above is very simple in this case. Namely, the CE parameters $\{(a_i, b_i^2)\}$ are updated via the sample mean and sample standard deviation of a fixed number of the highest scoring samples, so-called *elite* samples; i.e., those that give the highest likelihood. For the θ_i in a constrained region, that is for $i \ge 4k+1$, we sample from a *truncated* Gaussian distribution on the constrained region. It can be shown (Rubinstein and Kroese 2004) that the updating procedure is exactly the same as in the non-truncated case. For notational convenience we summarize the a_i and b_i^2 into vectors $\mathbf{a}$ and $\mathbf{b}^2$, and denote the Gaussian $\mathbb{R}^{6k-1}$-dimensional distribution with independent components with means $\mathbf{a}$ and variances $\mathbf{b}^2$ by $\mathsf{N}(\mathbf{a}, \mathbf{b}^2)$. The main algorithm is summarized as follows:

Algorithm 3.1 (CE Algorithm)

1. *Initialize* $\mathbf{a}_0$ *and* $\mathbf{b}_0^2$. *Set* $t = 1$ *(level counter).*

2. *Generate a sample* $\boldsymbol{\Theta}_1,\ldots,\boldsymbol{\Theta}_N$ *from* $\mathsf{N}(\mathbf{a}_{t-1}, \mathbf{b}_{t-1}^2)$ *(or its truncated version) and compute the log-likelihoods according to* (8).

3. *Let* $\tilde{\mathbf{a}}_t$ *and* $\tilde{\mathbf{b}}_t^2$ *be the sample means and variances based on the best* N^{elite} *samples.*

4. *Update the* $\mathbf{a}$ *and* $\mathbf{b}^2$ *in a "smooth" way as*

$$\begin{aligned} \mathbf{a}_t &= \alpha\,\tilde{\mathbf{a}}_t + (1-\alpha)\,\mathbf{a}_{t-1}, \\ \mathbf{b}_t^2 &= \alpha\,\tilde{\mathbf{b}}_t^2 + (1-\alpha)\,\mathbf{b}_{t-1}^2. \end{aligned} \qquad (9)$$

5. *Stop at iteration $t = T$ if some* **stopping criterion** *is met. Output* $\mathbf{a}_T$. *Otherwise increase t by 1 and return to step 2.*

The result is a sequence of parameters $(\mathbf{a}_0, \mathbf{b}_0^2)$, $(\mathbf{a}_1, \mathbf{b}_1^2), \ldots$ that tends to some $(\boldsymbol{\theta}^*, \mathbf{0})$ where $\boldsymbol{\theta}^*$ is the estimate of the global maximum. Note that during the course of Algorithm 3.1 the sampling distribution "shrinks" to a degenerate distribution. That is, each variance b_i^2 tends to 0, so that the mean a_i corresponds to the optimal θ_i.

The algorithm is quite robust under the choice of the initial parameters $\mathbf{a}_0$ and $\mathbf{b}_0^2$, provided that the initial variances are chosen large enough. A convenient choice is to let the initial means and variances be equal to the means and variances of the data.

Injection

When the smoothing parameter α is large, say 0.9, the convergence to a degenerate distribution may happen too quickly, which would "freeze" the algorithm in a sub-optimal solution. One way to prevent this from happening is to use *dynamic smoothing* (Rubinstein and Kroese 2004) where at iteration t the variance b^2 is updated using a smoothing parameter

$$\beta_t = \beta - \beta\left(1 - \frac{1}{t}\right)^q, \qquad (10)$$

where q is a small integer (typically between 5 and 10) and β is a large smoothing constant (typically between 0.8 and 0.99). The mean parameter a can be updated in the conventional way, with constant smoothing parameter α. By using β_t instead of α the convergence to the degenerate case has polynomial speed instead of exponential. A difficulty with dynamic smoothing is that when the optimal function value is unknown it is difficult to formulate a good stopping criterion due to the slower convergence of the algorithm.

In this paper we introduce a different technique, which can be applied to any optimization problem, and was observed to work very well for various multiextremal optimization problems. The idea is to "inject" extra variance into the sampling distribution in order to avoid premature shrinkage, according to the following recipe:

1. If during the course of Algorithm 3.1, at iteration t say, the maximum of the variances in $\mathbf{b}_t^2$ is less than ε (say 0.01), add
$$\left|S_t^* - S_{t-1}^*\right| h,$$
to the variances, for some h between 0.1 and 10. Here S_t^* is the best log-likelihood value obtained in iteration t.
2. If the number of variance injections exceeds some number d, say 5, then stop and display the best solution found, namely $\mathbf{a}_t$, otherwise increase t and proceed with the next iteration of Algorithm 3.1.

Running the variance injection method can be viewed as running the CE algorithm more than once.

Note that 2. above gives our stopping criterion in step 5 of the CE algorithm. Different stopping criteria are possible.

Another modification that proved useful in our situation was to update the means $\mathbf{a}$ and variances $\mathbf{b}^2$ using *different* smoothing parameters.

4 NUMERICAL EXPERIMENT

In this section we illustrate the performance of the EM and CE algorithms for a 2-dimensional clustering problem with 6 Gaussian clusters. It is not our intention to give here an exhaustive study, but we do believe that the present results give an indication of the usefulness of the CE method in comparison with the EM method.

For the EM algorithm we used the recent matlab implementation in the matlab *classification toolbox* (Stork and Yom-Tov 2004); see also (Duda, Hart, and Stork 2001). In the EM experiments we used "random" starting values as in (McLachlan and Krishnan 1997).

In the CE experiment we used a sample size of $N = 90$ and an elite sample size of $N^{\text{elite}} = 12$. Each mean $\mathbf{a}$ was updated using a smoothing parameter of 0.9. The variances were updated using a smoothing parameter of 0.3. The initial means and variances (in $\mathbf{a}_0$) were chosen equal to the mean and variance of the data. The initial weights (in $\mathbf{a}_0$) were chosen equal. The values in $\mathbf{b}_0^2$ were chosen large enough in order to provide a uniform sample from Ω in the first iteration. The injection parameters were $\varepsilon = 0.01$ and $h = 2$. Note that the CE method is fairly insensitive to the choice of the parameters. We stop the CE algorithm after $d = 5$ injections.

In the experiment $n = 200$ data points are drawn from a Gaussian mixture distribution with parameters given in Table 1.

Table 1: Parameters for the Experiment

$\boldsymbol{\mu}$	$\boldsymbol{\sigma}^2$	Cov	$\mathbf{w}$
(0.60,6.00)	(1.00,1.00)	0.90	0.10
(1.00,-10.00)	(1.00,1.00)	-0.90	0.10
(10.00,-1.00)	(2.00,2.00)	0.00	0.20
(0.00,10.00)	(2.00,2.00)	0.00	0.20
(1.00,-3.00)	(2.00,2.00)	-0.00	0.20
(-5.00,5.00)	(2.00,2.00)	0.00	0.20

The first column gives the mean vectors for the six Gaussian distributions in the mixture, the second column the variances, the third column the covariances and the last column the mixture weights.

To complete the specification of our test problem, we need to give the constraints θ^{low} and θ^{high}. For the present case we allow only variances greater or equal to 0.75, and correlation coefficients between -0.95 and 0.95.

Table 2 gives the evolution of a typical run of the CE algorithm for this data set, and the above constraints. In the table we list, from left to right, the best value for the *negative* of the log-likelihood in each iteration, the overall best value and the maximum CE variance. Note that we want to minimize the negative of the log-likelihood. For this run the CPU time – using a matlab implementation on a 2.4 GHz computer – was 38 seconds.

Table 2: Typical Evolution of the CE Algorithm

t	S^*_t	$\min_{u\leq t} S^*_u$	$\max_i b^2_t(i)$
20	1160.89	1142.00	35.30
40	1053.52	1052.65	24.62
60	1013.71	1013.38	3.01
80	1005.13	1005.13	0.98
100	1001.60	1001.55	0.91
120	998.45	998.45	0.06
140	1028.42	997.96	0.12
160	1002.02	997.96	0.05
180	998.81	997.96	0.03
200	997.00	997.00	0.02
220	1023.58	996.03	0.12
240	1000.91	996.03	0.19
260	990.64	990.64	0.08
280	982.11	982.11	0.02
300	1013.77	981.52	0.22
320	988.15	981.52	0.15
340	981.94	981.52	0.05
360	980.53	980.34	0.01
380	994.29	980.34	0.08
400	983.26	980.34	0.05
420	980.73	980.34	0.02
440	1015.13	980.34	0.21
460	985.78	980.34	0.13
480	981.75	980.34	0.04
500	980.43	980.34	0.01

In order to make a fair comparison we run multiple copies of the EM algorithm, so that the total time for all EM runs is no less than the time taken by the CE algorithm (38 seconds, here). Solutions of the EM algorithm that do not satisfy the constraints are discarded.

Table 3 gives the estimates using EM (using the best of all the multiple EM runs) and CE. In this case CE finds the global maximum but EM fails to do so. The difference in estimates is quite significant. Note that in this case EM finds only 5 clusters.

Table 3: Estimates for the Experiment

	$\boldsymbol{\mu}$	$\boldsymbol{\sigma}^2$	Cov	**w**
EM	(4.07, 4.34)	(2.12, 1.75)	-1.73	0.09
	(5.16 5.94)	(1.00, 1.58)	-0.51	0.10
	(10.11, -0.94)	(2.61, 1.90)	0.26	0.20
	(1.09, -5.01)	(1.79 14.678)	-0.06	0.31
	(0.25, 8.73)	(1.97, 5.25)	0.20	0.29
	(0, 0)	(1 , 1)	0	0
CE	(0.24, 5.91)	(0.81,0.75)	0.72	0.08
	(1.05 , -10.01)	(0.79,0.76)	-0.71	0.11
	(10.10, -0.90)	(2.52,1.98)	0.14	0.21
	(0.07, 9.97)	(2.31,2.17)	0.53	0.19
	(1.11, -2.78)	(2.44,1.67)	0.25	0.20
	(-4.45 ,5.25)	(1.88,2.66)	-0.55	0.21

Figure 1 illustrates the quality of the estimates. The 0.95-quantile ellipses found with CE correspond exactly with the data, whereas EM only “recognizes” one cluster.

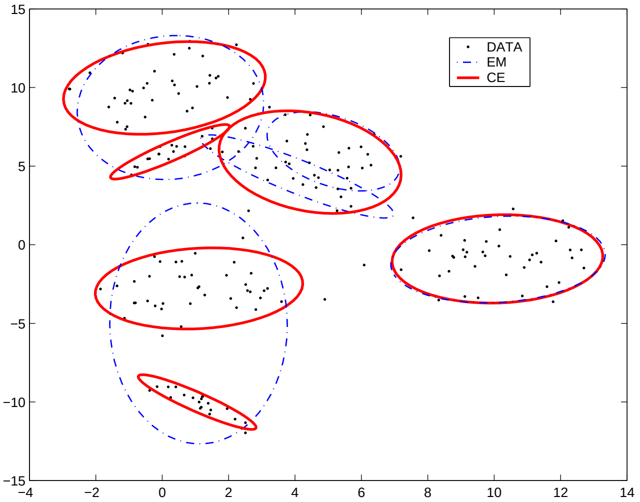

Figure 1: Illustration of the Experiment

We have repeated this experiment 10 times, in which CE found the correct clusters 4 out of 10 times, but EM failed. The results are given in Table 4. Note that here, again, the negative of the log-likelihood values are given.

Table 4: The Negative of the Log-Likelihood Values for CE and EM for 10 Repetitions

CE	EM	time
980.33	1048.62	38
982.75	1052.99	32
998.01	1041.86	34
980.36	1047.83	31
980.32	1047.83	36
997.98	1052.99	34
994.19	1047.83	40
1004.76	1057.18	39
994.08	1052.99	35
980.62	1052.99	34

It is important to note that the CE method is not sensitive to the initial conditions. This is in sharp contrast with the performance of the EM method, the convergence of which is heavily dependent on an initial guess that is not too far away from the optimal solution. This explains why the CE method appears more consistent than the EM method.

5 CONCLUSIONS AND FUTURE RESEARCH

We have introduced a constrained global likelihood optimization approach to (mixture) model-based clustering analysis, based on the CE method. Numerical experiments indicate that the new method is very effective and could be used as an alternative to the ubiquitous EM algorithm. When comparing the two, the main advantage of EM is its speed. We observed that, typically, EM converges 10–100 times faster than CE. However, this was using a state-of-the art implementation for EM and a non-optimized implementation for CE. The disadvantages of EM are that (1) EM requires "correct" starting values and (2) it is difficult to deal with constrained parameters, other than just accepting or rejecting a candidate solution generated by EM.

The advantages of CE is that it deals better with both starting values and constraints. The differences between CE and EM become more clear when the data set is small to medium, 20–300, and the clusters are superimposed. In that case the log-likelihood function has many local maxima.

Under what conditions exactly CE outperforms EM in clustering analysis remains an issue for future research and requires more numerical experimentation. Moreover, there are various other approaches to clustering analysis, including the *K-means* and the *linear vector quantization* algorithms (Duda, Hart, and Stork 2001). Another direction for future research is to compare CE with these algorithms. Early results indicate the superior accuracy of CE.

One of advantages of the CE method which we have not mentioned yet is that it can be readily applied to other multi-extremal optimization problems. The core code remains virtually the same; only the objective function needs changing. A study on the performance of the CE method, when applied to a whole range of difficult optimization problems, be it likelihood optimization or otherwise, is an interesting direction for future research. Another issue worth considering is the "optimal" choice of the CE parameters. As we mentioned, the algorithm is fairly insensitive to the choice of parameters, but additional information about for example the selection of the injection parameter h would be useful. Other modifications for clustering problems include the "gradual feeding" of data, where one starts with say 20% of the data to identify the clusters quickly and then gradually increases the data set to determine the actual clusters. Also a different choice of the sampling distribution may be considered. Examples are the class of beta distributions and the class of double exponential distributions.

ACKNOWLEDGMENT

We thank Uri Dubin for directing us to the EM algorithm in the matlab *classification toolkit*, and Thomas Taimre for helpful comments.

REFERENCES

Alon, G., D. P. Kroese, T. Raviv, and R. Y. Rubinstein. 2004. Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment. *Annals of Operations Research*. To appear.

Boyles, R. A. 1983. On the convergence of the EM algorithm. *Journal of the Royal Statistical Society B* 45 (1): 47–50.

Chepuri, K., and T. H. de Mello. 2004. Solving the vehicle routing problem with stochastic demands using the cross entropy method. *Annals of Operations Research*. To appear.

de Boer, P. T., D. P. Kroese, S. Mannor, and R. Y. Rubinstein. 2004. A tutorial on the cross-entropy method. *Annals of Operations Research*. To appear.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* 39 (1): 1 – 38.

Dubin, U. 2002, June. The cross-entropy method for combinatorial optimization with applications. Master's thesis, The Technion, Israel Institute of Technology, Haifa.

Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern classification*. Wiley, New York.

Goldberg, D. 1989. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley.

Helvik, B. E., and O. Wittner. 2001. Using the cross-entropy method to guide/govern mobile agent's path finding in networks. In *3rd International Workshop on Mobile Agents for Telecommunication Applications - MATA'01*.

Liu, Z., A. Doucet, and S. S. Singh. 2004. The cross-entropy method for blind multiuser detection. In *IEEE International Symposium on Information Theory*. Chicago.

Mannor, S., R. Y. Rubinstein, and Y. Gat. 2003, August. The cross-entropy method for fast policy search. In *The 20th International Conference on Machine Learning (ICML-2003)*. Washington, DC.

Margolin, L. 2002, July. Cross-entropy method for combinatorial optimization. Master's thesis, The Technion, Israel Institute of Technology, Haifa.

Margolin, L. 2004. On the convergence of the cross-entropy method. *Annals of Operations Research*. To appear.

McLachlan, G., and T. Krishnan. 1997. *The EM algorithm and extensions*. John Wiley & Sons.

Nelder, J. A., and R. Mead. 1965. A simplex method for function minimization. *Comput. J.* 7:308–313.

O. Wittner and B. E. Helvik 2002, May 12-17th. Cross Entropy Guided Ant-like Agents Finding Dependable Primary/Backup Path Patterns in Networks. In *Proceedings of Congress on Evolutionary Computation (CEC2002)*. Honolulu, Hawaii: IEEE.

Rubinstein, R. Y. 1999. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability* 2:127–190.

Rubinstein, R. Y. 2001. Combinatorial optimization via cross- entropy. In *Encyclopedia of Operations Research and Management Sciences*, ed. S. Gass and C. Harris, 102–106: Kluwer.

Rubinstein, R. Y. 2002. The cross-entropy method and rare-events for maximal cut and bipartition problems. *ACM Transactions on Modelling and Computer Simulation* 12 (1): 27–53.

Rubinstein, R. Y., and D. P. Kroese. 2004. *The cross-entropy method: A unified approach to combinatorial optimization, monte-carlo simulation and machine learning*. New York: Springer-Verlag.

Stork, D. G., and E. Yom-Tov. 2004. *Computer manual to accompany pattern classification*. Wiley.

Wu, C. F. 1983. On the convergence properties of the EM algorithm. *The Annals of Statistics* 11 (1): 95–103.

AUTHOR BIOGRAPHY

ZDRAVKO BOTEV is currently pursuing a B.Sc. degree in Mathematics and Statistics at the University of Queensland. He has won several awards, including the competitive Australian Bureau of Statistics scholarship in Mathematical Statistics. He is expected to complete his Science baccalaureate at the end of 2004.

DIRK P. KROESE has written close to 40 papers in a wide range of topics in applied probability and simulation. He is a pioneer of the well-known *cross-entropy* method and co-author (with R.Y. Rubinstein) of the first monograph on this method. He is associate editor of *Methodology and Computing in Applied Probability* and guest editor of *Annals of Operations Research*. He has held research and teaching positions at Princeton University and the University of Melbourne, and is currently working at the Department of Mathematics of the University of Queensland, where he enjoys teaching and researching stochastic modeling, applied probability and statistics. His web address is `http://www.maths.uq.edu.au/~kroese/`