Improved classification em algorithm for the problem of separating semiconductor device production batches

Victor Orlov[1], Lev Kazakovtsev[1,2], Ivan Rozhnov[1], Ekaterina Bezhitskaya, Sergey Bezhitskiy
Reshetnev Siberian State University of Science and Technology,
prosp. Krasnoyarskiy Rabochiy 31, Krasnoyarsk 660031, Russia
Siberian Federal University, 79 Svobodny av., Krasnoyarsk, 660041, Russia
levk@bk.ru

Abstract.
This paper focuses on new proposed algorithms for cluster problem solving. The proposed algorithms is based on Classification EM algorithm (CM-algorithm). The algorithms are new algorithms of the greedy heuristic method using the idea of searching in alternating neighborhoods. The numerical experiments shows that the proposed algorithms have less mean values and/or less standard deviation of objective function, less scatter of obtained values in comparison with classical CEM-algorithm.

1. Introduction

There are a lot of methods for classifying and clustering data [1, 2]. One of the most popular methods includes the EM algorithm (Exception Maximization). The EM algorithm is successfully used for statistical tasks related to the analysis of incomplete data (if some statistical data are missing due to some reason) or for cases when the likelihood function has a form that does not allow "convenient" research methods, but allows serious simplifications when introducing additional "unobservable" ("hidden") values [1, 3].

The separating problem of radio and radio products homogeneous production batches is precisely the normal distribution multidimensional data clustering problem with hidden data [4].

2. EM-algorithm

The distribution density on the set X has the form of a mixture of k distributions (we assume that the distributions are Gaussian) [5]:

$$\rho(x) = \sum_{j=1}^{k} \alpha_j \rho_j(x), \ \sum_{j=1}^{k} \alpha_j = 1, \ \alpha_j \geq 0,$$

here $\rho_j(x)$ - is the likelihood function of the *j*-th component of the mixture , α_j - is its prior probability.

Let the likelihood functions belong to a parametric set of distributions $\varphi(x;\theta)$ and differ in the parameter values only $\rho_j(x) = \varphi(x;\theta_j)$.

The fuzzy clustering problem (separation of the mixture) is to estimate the vector of parameters $\Theta = (\alpha_1,..., \alpha_k, \theta_1,..., \theta_k)$ with a sample X^m of random and independent observations from the mixture $\rho(x)$, knowing the number k and the function φ.

The idea of the expectation maximizing algorithm is to artificially introduce an auxiliary vector of hidden variables G with the following features:

- it can be calculated if the values of the parameter vector are known Θ;
- if the values of hidden variables are known then the search for maximum likelihood is greatly simplified.

This allows us to transform a complex optimization problem to a sequence of iterations for recalculating coefficients (hidden variables calculation in the current approximation of the parameter vector — E-step) and maximizing the likelihood (it is necessary to find the next approximation of the vector — M-step).

The vector component values G of the hidden variables are calculated from the current approximation of the parameter vector Θ. Let us denote $\rho(x;\theta_j)$ as probability density. The $\rho(x;\theta_j)$ means that the object x is derived from the j-th component of the mixture using the formula of conditional probability: $\rho(x,\theta_j) = \rho(x)P(\theta_j \mid x) = \alpha_j \rho_j(x)$.

Let us take $g_{ij} \equiv P(\theta_j \mid x_i)$. This is the unknown a posteriori probability. The $g_{ij} \equiv P(\theta_j \mid x_i)$ pobability means that the training object x_i is obtained from the j-th component of the mixture. These values are used as hidden variables. The $\sum_{j=1}^{k} g_{ij} = 1$, for anyone $i = 1,..,m$ because it makes sense for the total probability to belong to the object x_i of one of the k components of the mixture.

The likelihood maximizing problem is being solving at the M-step. And then the next approximation of vector Θ is being finding at M-step using current values of the vectors G and Θ.

The log likelihood maximizing problem statement:

$$Q(\Theta) = \ln \prod_{i=1}^{m} \rho(x_i) = \sum_{i=1}^{m} \ln \sum_{j=1}^{k} \alpha_j \rho_j(x_i) \to \max_{\Theta}$$

Having solved the Lagrange optimization problem with constraint α_j, we find:

$$\alpha_j = \frac{1}{m} \sum_{i=1}^{m} g_{ij},\ j = 1,..,k \qquad \theta_j = \arg\max_{\theta} \sum_{i=1}^{m} g_{ij} \ln \varphi(x_i,\theta),\ j = 1,..,k$$

Therefore, the M-step consists of k independent optimization problems. The weights of the components α_j are being calculating as arithmetic averages at M-step. The parameters θ_j are being estimating at M-step. It is worth noting that the separation of variables was made possible by the introduction of hidden variables.

The iteration process stops according to a previously agreed stopping criterion (a pre-selected metric $\rho(\theta_1,\theta_2)$ and number ε). The process stops at the m-th step, if $\rho(\theta^{(m)},\theta^{(m-1)}) < \varepsilon$.

3. Modification of the EM-algorithm

In practice, the algorithm based on the method of greedy heuristics and the k-means model [6] is used for dividing a sample of electronic components into homogeneous production lots. This algorithm does not allow determining the number *k* (number of clusters) of the components of the mixture. The *k* value should be set before the start of the algorithm or a series of problems with different estimated number of clusters should be solved

It was established experimentally [5] that the EM-algorithm results has a strong dependence on the initial data. For example, replacing one only observation with another can drastically change the final

estimates that obtained using the EM algorithm in the case of a four-component mixture of normal laws with a sample size of 200–300 observations [3]. The median modifications of the basic EM algorithm are proposed to exclude this kind of instability [1].

Modification of the classical EM algorithm [1] is the Classification EM algorithm (CEM algorithm). The CEM algorithm works according to the principle of clear classification of sample data. The CEM modification of the algorithm coincides with SEM (Stochastic EM) but has specific feature. The specific feature is to use a deterministic rule at each step. The deterministic rule is to put data into one of the clusters that has maximum of calculated posteriori probability.

1. The idea of a new algorithms based on the CEM algorithm

The idea of presented paper is to apply the greedy heuristics method for the automatic grouping problem solving.

Given: There are two known (parent) problem solutions. Solutions are represented by set pairs <*D, W*>. The set *D* is the set of distributions in the mixture. Each distribution is given by parameters. The second set W is the prior probabilities corresponding to the distributions.

Algorithm 1. Basic greedy agglomerative heuristics.

Given: the initial number of clusters *K*, the required number of clusters *k*. Moreover, K > k.

1. Choose an initial solution with *K* clusters, i.e. randomly select the initial parameters of a pair of sets of distributions and their weights $< D,W >=< \left\{N(\mu_i^{\langle 0\rangle}, {\sigma_i}^{\langle 0\rangle 2} I_n)\right\}, \left\{\alpha_i^{\langle 0\rangle}\right\}, i = \overline{1,K} >$.

2. Run the EM algorithm and get a new (improved) solution represented by <*D, W*>.

3. If $K = k$, then stop.

4. For each $i' \in \left\{\overline{1,K}\right\}$ perform:

4.1. Get a pair of truncated sets $< D',W' >=< D \setminus \left\{N(\mu_{i'}^{\langle 0\rangle}, {\sigma_{i'}}^{\langle 0\rangle 2} I_n)\right\}, \quad W \setminus \{\alpha_{i'}^{\langle 0\rangle}\} >$.

4.2. Run the EM algorithm with the initial values of the distribution parameters represented by the truncated <*D ', W'*>. In this case, the EM algorithm is limited to one iteration. Then, it is necessary to calculate the objective function *L* for each solution that is obtained using EM algorithm. Save the calculated value of *L* function as $L'_{i'}$

4.3. Next iteration of loop 4.

5. Find the index $i'' = \arg\max_{i'=\overline{1,k}} L_{i'}$.

6. Obtain a pair of truncated sets $< D'',W'' >=< D \setminus \left\{N(\mu_{i''}^{\langle 0\rangle}, {\sigma_{i''}}^{\langle 0\rangle 2} I_n)\right\}, \quad W \setminus \{\alpha_{i''}^{\langle 0\rangle}\} >$. Run the EM algorithm for obtained pair of truncated sets, then go to step 3.

In this paper, the new heuristic procedures are proposed.

Algorithm 2. A greedy procedure with partial merger No.1.

Given: a pair of sets of distributions $< D',W' >=< \left\{N(\mu'^{\langle 0\rangle}_i, {\sigma'}^{\langle 0\rangle 2}_i I_n)\right\}, \left\{\alpha'^{\langle 0\rangle}_i\right\}, i = \overline{1,K} >$ and $< D'',W'' >=< \left\{N(\mu''^{\langle 0\rangle}_i, {\sigma''}^{\langle 0\rangle 2}_i I_n)\right\}, \left\{\alpha''^{\langle 0\rangle}_i\right\}, i = \overline{1,K} >$

1. For each $i' \in \left\{\overline{1,k}\right\}$ perform:

1.1. Merge element by element sets in pairs <*D ', W'*> and <*D '', W ''*> $< D,W >=< D' \cup \left\{N(\mu''^{\langle 0\rangle}_{i'}, {\sigma''}^{\langle 0\rangle 2}_{i'} I_n)\right\}, W' \cup \{\alpha'^{\langle 0\rangle}_{i'}\} >$.:

1.2. Run the basic greedy heuristics (Algorithm 1) with these pairs of merged sets <*D, W*> as the initial solution. Save the obtained result (a pair of the resulting sets, as well as the value of the objective function).

3. The best obtained solution in step 1.2 returns as a result.

Algorithm 3. A greedy procedure with a complete union of parental decisions.

Given: see Algorithm 2

1. Merge element by element sets <*D ', W'*> and <*D '', W ''*>: $< D,W >=< D'\cup D'',\ \ W'\cup W'' >$.

2. Run Algorithm 1 with merged sets. Take the merged sets as the initial solution.

Algorithm 4. A greedy procedure with a partial merger № 2.

Given: see Algorithm 2

1. Choose a random value $r \in \overline{\{2,k-1\}}$ with equal probability.

2. Repeat k-r times:

2.1. Form a randomly selected subset *D'''* from the elements of the *D''* set. The power of *D'''* subset is *r*. Form subset *W'''* from elements of the W'' set (the power of *W'''* is equal r).

2.2. Merge sets $< D,W >=< D'\cup D''',\ \ W'\cup W''' >$.

2.3. Run Algorithm 1 with merged sets as the initial solution.

3. Return the best (by the value of the objective function) solution from the solutions obtained in step 2.3 as a result.

The first computational experiments showed the extreme inefficiency of Algorithm 4 in comparison with Algorithm 3 for all the solved problems [4]. At the same time, efficiency is greatly increased if the number of elements added from the solution <*D'', W''*> is limited as follows:

Step 1 of Algorithm 4: Select random $r' \in [0;1)$. Assign r = [(k / 2-2) r'2] +2. Here [.] is the integer part of the number.

Proposed heuristic procedures are (not in the strict sense) local search algorithms in the neighborhood of a well-known ("parent") solution that is represented by <*D', W'*>. These proposed procedures can be used as part of various global search strategies. At the same time, the neighborhoods are ("children") solutions that are derived from <*D ', W'*> solution that is merged with the elements of <*D'', W''*> solution and using basic greedy agglomerative heuristic (Algorithm 1).

The EM algorithm and its modifications (CEM, SEM, median EM [1, 7]) can be used quite successfully as a local search method. On the one hand, the global search algorithm should periodically "knock out" an intermediate solution of the problem from the "region of attraction" of the local optimum. On the other hand, solutions formed from elements of various local-optimal solutions are more likely to be closer to the global optimum in comparison with randomly selected solutions [8]. Thus, it seems promising to search in a neighborhood of a local optimum. The local optimum neighborhood is formed by replacing individual elements of a local-optimal solution with elements of other local-optimal solutions. Such approach is used in different variants of Algorithm 4. The intermediate solutions of Algorithm 4 are represented by pairs of sets of distributions and their weights. Each of set pair is the result of the EM-algorithm. That is the local maximum. Therefore, it is proposed in this paper to use the VNS algorithm as an extended local search [9, 10].

A search in the surroundings formed by adding to the well-known intermediate solution represented by a pair of sets <D ', W'> of elements of another solution <D '', W ''> with the subsequent removal of the "extra" solutions by greedy agglomerative heuristic procedure is performed by Algorithms 2, 3 and 4. Thus, these algorithms search in some neighborhoods of the solution <*D ', W'*>, and the second solution <*D'', W''*> implicitly sets the parameters of this neighborhood. Thus, Algorithm 2 searches in the neighborhood of *S (*<*D', W'*>*)*$=\{greedy(< D',W' > \cup \{< D''_i, W''_i >\}), i = 1, |D''|\}$. Here *greedy ()* is the result of Algorithm 1 applying. Accordingly, Algorithms 3 and 4 search in wider neighborhoods.

Thus, the general scheme of the search algorithm in alternating neighborhoods can be described as follows:

Algorithm 5. CEM-VNS

1. Run the CEM-algorithm from a random initial solution and get the solution <*D, W*>.

2. Set *s*= s_{start} (it is a № of search neighborhood)

3. Set *i=0, j=0;* (the quantity of unsuccessful iterations in a particular neighborhood and as a whole by the algorithm).

4. Run the CEM algorithm from a random initial solution, get the solution *<D', W'>*.

5. Depending on the value of *s* (values of 1, 2 or 3 are possible), run Algorithm 2, 3 or 4 with the initial solutions *<D, W>* and *<D', W'>*. Thus, a neighborhood is determined by the procedure for including distributions from the second known solution (the mentioned Algorithms 2, 3, or 4) and the neighborhood parameter that is the second known solution. The search is carried out in this particular neighborhood.

6. If the result value of the objective function is better than *<D, W>*, then replace *<D, W>* with this new result, assign *i = 0, j = 0*, go to Step 5.

7. Assign i = i + 1;

8. If *i <imax*, then go to Step 4.

9. Assign i = 0, j = j + 1. Make the transition to a new neighborhood: *s = s + 1*; if s> 3, then assign *s = 1*;

10. If *j> jmax*, or other stop conditions are satisfied (for example, the maximum running time), then STOP. Otherwise, go to Step 5.

The s_{start} parameter specifies the number of the neighborhood. This number is index of start neighborhood from which the search begins, is particularly important [10]. We performed computational experiments with all its possible values. Depending on this value, the algorithms are designated below, respectively, CEM-VNS1, CEM-VNS2, CEM-VNS3. The start of the search algorithm can run from different neighborhoods.

1. Computational experiment results

The following abbreviations and abbreviations of algorithms are used: CEM - Classification EM algorithm; CEM-VNS1, CEM-VNS2 and CEM-VNS3 are variants of the search algorithm in alternating neighborhoods.

As test data sets, the results of non-destructive test experiments of prefabricated production batches of radio products are researched (table 1). These experiments are performed in a specialized test center to complete the onboard equipment of spacecraft.

The DEXP OEM computing system (4-core Intel® Core ™ i5-7400 CPU 3.00 GHz, 8 GB of RAM) is used for researching of proposed algorithms.

For all data sets, 30 attempts are made to run each algorithm. Only the best obtained results in each attempt are recorded. The objective function values: minimum value (Min), average value (Average) and standard deviation (RMS) were calculated for each algorithm using the best obtained results.

Table 1. Computational experiment results for data set of radio products (10 clusters, 2 minutes, 30 attempts)

Algorithm	Value of objective function			
	Min	Max	Mean	SD
3OT122A (767 vectors of data, each vector has 13 component)				
CEM	120 947.6	146 428.5	135 777.6	***7 985.6992***
CEM-VNS1	108 979.8	152 729.1	141 728.3	11 421.9262
CEM-VNS2	123 664.4	158 759.2	***143 028.5***	10 294.3992
CEM-VNS3	***128 282.2***	155 761.9	***143 506.9***	10 058.8266
1526TL1 (1234 vectors of data, each vector has 157 component)				
CEM	354 007.3	416 538.4	384 883.4	***20 792.8068***
CEM-VNS1	376 137.1	477 124.5	438 109.4	29 964.0641
CEM-VNS2	345 072.6	487 498.3	444 378.1	43 575.3282
CEM-VNS3	***379 352.3***	516 777.8	***456 271.4***	38 323.0246

2. Conclusion

The computational experiments show that the stability of the results with multiple launches of the CEM algorithm is higher than the EM algorithm has. At the same time, the result is in many cases is far from the true likelihood function optimum. In general it is practically impossible to determine the true optimum for large problems. The analysis of results shows that there exists the available reserve for result improving. This reserve can be explained by enough bigger difference between value of the best attempts and mean value for both the EM algorithm and its modifications.

The results of performed computational experiments show that new proposed search algorithms in alternating neighborhoods (CEM-VNS) have more stable results (give a lower mean value and / or a standard deviation of the objective function, a smaller scatter of the achieved values) and, consequently, better performance in comparison with the classical CEM -algorithm. The comparative effectiveness of new proposed algorithms on various data sets has been experimentally proven.

3. Acknowledgements

Results were obtained in the framework of the state task No. 2.5527.2017/8.9 of the Ministry of Education and Science of the Russian Federation.

References

[1] Korolev V 2007 EM-algorithm, its modications and their application to the problem of separation of mixtures of probability distributions *Theoretical review* IPIRAN Moscow.

[2] Cherezov D and Tyukachev N 2009 Overview of basic data classification and clustering methods *Bulletin "System Analysis and Information Technologies"* **2** Voronezh.

[3] Kazakovtsev L, Orlov V and Stupina A 2015 The choice of a metric for a system for the automatic classification of radio products by production batches *Software products and systems* **2** 124–29 Doi: 10.15827/0236-235X.110.124-129.

[4] Kazakovtsev L, Orlov V, Stashkov D, Antamoshkin A and Masich I 2017 Improved model for detection of homogeneous production batches of electronic components *IOP Conference Series: Materials Science and Engineering* **255** doi:10.1088/1757-899X/255/1/012004.

[5] Kazakovtsev L, Stashkov D, Gudyma M and Kazakovtsev V 2019 Algorithms with Greedy Heuristic Procedures for Mixture Probability Distribution Separation *Yugoslav Journal of Operations Research* **29** 51-67

[6] Kazakovtsev L 2016 The greedy heuristics method for systems of automatic grouping of objects Diss ... Dr. tech. of science Krasnoyarsk.

[7] Celeux G and Diebolt J 1984 Reconnaissance de m´elanges de densit´e et classification *Un algorithme d'apprentisage probabiliste: l'algorithme SEM* Rapport de Recherche de l'INRIA RR-0349 Centre de Rocquencourt.

[8] Celeux G and Govaert A 1991 Classification EM Algorithm for Clustering and Two Stochastic Versions *Rapport de Recherche de l'INRIA* RR-1364 Centrede Rocquencourt.

[9] Kazakovtsev L, Antamoshkin A and Masich I 2015 Fast Deterministic Algorithm for EEE Components Classification *IOP Conf. Series: Materials Science and Engineering* **94** Article ID 012015, 10 P. DOI: 10.1088/1757-899X/04/1012015.

[10] Orlov V, Kazakovtsev L, Rozhnov I, Popov N and Fedosov V 2018 Variable neighbourhood search algorithm for k-means clustering *IOP Conf. Series: Materials Science and Engineering* **450** Article ID 022035, DOI:10.1088/1757-899X/450/2/022035.