ISSN 1870-4069

Robust Algorithm of Clustering for the Detection of Hidden Variables in Bayesian Networks

Niels Martínez-Guevara, Nicandro Cruz-Ramírez, José-Rafael Rojano-Cáceres

Universidad Veracruzana,
Facultad de Estadística e Informática, Mexico

{niemartinez,ncruz,rrojano}@uv.mx
http://www.uv.mx

Abstract. In machine learning there are tree principal areas where the computer is able to learn, the first one is supervised learning, the second one is unsupervised learning and the third one learning by reinforcement. On the first appears something that the literature calls the "learning problem" where they study the most common scenes in relationship to the data and the model, in this paper we purpose a modification of the MS-EM algorithm that estimated a hidden variable from incomplete data in a scene where the model is unknown, and this algorithm is able to get a Bayesian network that explains the presence and relationship of the hidden variable and the dataset.

Keywords. MS-REM, MS-EM, clustering, likelihood, MDL.

1 Background and Introduction

Bayesian Networks or Belief Networks (BN) are graphical representations for probability distribution. They are defined by two components, on the one hand a directed acyclic graph where each node represent a variable of the dataset, and the arc represents a conditional independence between variables. On the other hand, there is a collection of *local interaction models* that describe the conditional probability of each variable [9].

Drawing a Bayesian network from experts can be an expensive process in large applications, however it is also possible to get it from data. Unfortunately the current methods are essentially limited to learn parameters for a fixed network structure [6].

Thus, in order to get a Bayesian Network it is possible to use the Model Selection Expectation-Maximization Algorithm, which also it is capable to obtain hidden values from data. To introduce the concept of hidden values we can take a example given by Heckerman; supposing we have a dataset from a sale cars company, where it is stored information about the last sales, we want to know the economic status of the clients but such information is not implicit in the data. However there are two variables with a big impact to estimated the hidden

variable: The price of the car and the monthly payment of the car. Thus, the social economic status of the client can be inferred as better if the price of the car and the monthly payment are high, in other case if the price of the car and monthly payment are low the social economical status is worst [8].

In this paper we analyze the MS-EM algorithm a variant of the clustering algorithm EM that proposes finding an alternative to solve two problems that arise during the process of clustering. The first one is how many values can take a hidden variable? In the special case of EM one of its virtues can also become a big problem, being a soft-clustering algorithm the number of clusters is dynamic and the traditional version of this algorithm obtained his parameters randomly in a range of values with a normal distribution $2 \leq c \leq n$ where c is the number of clusters and n the number of cases in the database, allowing in the first instance the case of duplicate clusters [4].

From a first random experimentation, the algorithm will adjust its values to find the values of the variable, the initial number of clusters c is a random value within a range, the results of the algorithm will be variable and it will be difficult to determine the number of possible values that variable can take since it presents us in the first instance two scenarios, the first, being the latent variable a variable with two possible values ??or the other end being each case of the data set pertaining to a cluster. This can be a big problem because once the initial number of clusters is defined, it can be reduced, however it can not be increased, altering the behavior of the algorithm and generating noise in the execution of the algorithm [11].

The proposal of MS-REM is that the numbers of clusters can be obtained from more to less by adjusting their number until it is adequate, while finding a network that maximizes the value of MDL, implementing an adaptive algorithm such as REM (Robust Expectation-Maximization) which is updating its values? and the number of clusters depending on the iterations of the algorithm, adjusting its probabilities to have a better Log-Likelihood.On the other hand, the original purpose of MS-EM algorithm takes the ability to see the relationship of the variable with the proposed model. In this way, we intend to implement a technique for obtaining the number of clusters and their model.

2 Analysis of the MS-EM Algorithm

Now, the proposal offered by the literature is as follows, if a Bayesian network could be found that explains the data and shows the presence of a hidden variable, as well as the relationship with the observed data, the two problems would be solved. The proposed algorithm is as follows [7]:

```
Procedure MS-EM:
    Choose M(0) and params(0) randomly
        Loop for n=0,1,... until convergence
        Find a Model Mn+1 that maximize Q(:Mn, params n)
        let params n+1 = arg max Q(Mn+1, params : Mn, params n)
```

where M is a Bayesian network and parameters is the probability distribution of the hidden variable. Now the algorithm shows only two steps, so next we will analyze the Expectation function where is represented by the formula 1 and it's the maximum likelihood function [10]:

$$argmaxQ = P(H|M, \theta). \tag{1}$$

Having an initial model M and a random initialization of clusters, we seek to obtain the parameters or the probability θ that each tuple belongs to a cluster H. The traditional version of EM indicates that the number of clusters will be calculated randomly and will be a number between $2 < c < n$ and the probability of each cluster is assigned randomly according to the initial experiment. In each iteration the probabilities of the clusters will be adjusted until there is no significant change, which can be defined in the error variable ε, which is the termination condition of the algorithm [5].

The next step is maximization "Find a Model" Finding a Bayesian network is a problem with high computational complexity, there is a considerable number of possible networks that can represent a database, so to obtain a solution a search algorithm was implemented that was able to find a "better structure" that represent better the relationships between the variables seen in the observe data. To do this we use the Random Mutation Hill-Climbing algorithm (RMHC). And to compare two or more Bayesian structures, the MS-EM algorithm uses the MDL (Minimum Description Length) metric which is describe in the formula 2 [2]:

$$MDL(B_S, D) = -N \cdot H(B_S, D) - \frac{1}{2}K \cdot \log N, \tag{2}$$

where $K = \sum_{i=1}^{n} q_i \cdot (r_i - 1)$ y $H(B_S, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}}$.

This formula expresses the following: N is the total number of cases, $H(B_S, D)$ is the Likelihood and $-\frac{1}{2}K \cdot \log N$ is the penalty value where preference is given to networks with fewer arcs and where the value q_i is referring to the number of combinations of the values of variables that can be made between a parent and child node, in the cases of N_{ijk} and N_{ij} referring to the formula are joint frequencies of the parent nodes with respect to the children, which are analysed in the database D, given the structure B_S [2].

Through this formula a negative value will be obtained that the closer it is to zero gets a better is the fitness, following the Ockham's razor principle the better network it's the simples network that represents better the data [3].

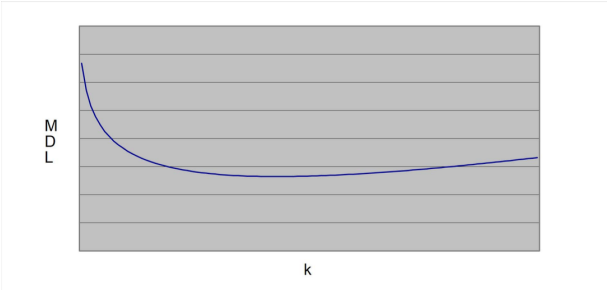

Fig. 1. MDL graph of behavior [3].

With all the formulas and metrics explained, the algorithm can be developed as follows:

1. Step 1: Randomly generate a model and a probability distribution, where the number of clusters is a random value between $2 < c < n$ and the sum of the mean of each cluster is equal to 1.
2. Step 2: Obtain a new network that improves the value of MDL through a RMHC search algorithm.
3. Step 3: Obtain a new probability distribution from the Maximum Likelihood with the new network.
4. Step 4: The two Bayesian networks are compared according to the MDL metric.
5. Step 5: if $(New\_MDL - Current\_MDL! = 0)$ AND $(t < 30)$then
   (a) The new network replaces the current one and returns to Step 2.
6. Else: Return the best distribution of probability and best network.

The previous algorithm is a variant of the EM algorithm where we try to find the "parameters" that we will call Probability Function (Maximum Likelihood) and the structure of the data (Bayesian Network) through the search algorithm.

However, certain limitations that the clustering algorithm has when calculating the number of clusters are still present. One of the proposals to solve this problem is the algorithm called Robust Expectation-Maximization [7].

3 MS-REM Algorithm Construction

During the design phase for the proposal that unites the ideas of an algorithm that focuses on obtaining a categorization from its model and a robust clustering algorithm, it was possible to design the MS-REM algorithm (Model Selection Robust Expectation Maximization) where the characteristics offered by the literature with respect to this algorithm were added, most of these modifications are focused on the calculation of clusters since in their initial configuration, the number of them is no longer a random value between 0 and n, but starts with an initial value of n, where n is the number of cases observables.

The algorithm starts taking into account that it is in the worst case and from it generates a random experiment, where each cluster has a probability of appearing $1/n$ in this first iteration it is possible that some media clusters have a probability of appearing equal to 0 and in other cases they will increase, thus generating larger clusters. This is why the adjustment functions of the algorithm were optimized to eliminate the smallest clusters (the literature suggests eliminating clusters whose mean is less than $1/n$) encouraging that larger clusters have a greater probability of appearing in the next iterations.

Although the iterative process appears to be slower in the first instance, the algorithm starts from the most general to the specific, thus encouraging the data to determine the appropriate groupings for each data set. However, this also has its weaknesses, since in subsequent investigations it was possible to verify in data generated by artificial distributions, that the algorithm allows small clusters, thus generating noise in the classification process cite soor.

With those changes in mind in the MS-REM clustering algorithm is presented as follows:

1. Step 1: Randomly generate a model and a probability distribution, where the **number of clusters is n** and the sum of the mean of each cluster is equal to 1.
2. Step 2: **The number of Clusters is adjusted, eliminating those clusters whose mean is less than 1 / n and the clusters are re-named.**
3. Step 3: Obtain a new network that improves the value of MDL, through the RMHC search algorithm.
4. Step 4: Obtain a new probability distribution from the Maximum Likelihood with the new structure.
5. Step 5: The two Bayesian networks are compared according to the MDL value.
   if $(New_MDL - Current_MDL == 0)$then
   Returns The best probability distribution and best network.
6. Else:
   - if $t < 60$ then: **The network is replaced and returns to Step 2.**
   - Else: **The network is replaced and returns to Step 3.**

When initializing with an initial number of clusters n, it is necessary to make a double adjustment in the algorithm, the first to determine the number of clusters and the second to determine its size.

4 Experimental Results

4.1 Tests with Synthetic Databases

In a first stage of experimentation, two Bayesian networks were defined 3x1x3 (figure 2a) and 3x2x4 (figure 2b) with five random probability distributions and different sample sizes (250,500,1000,2000).

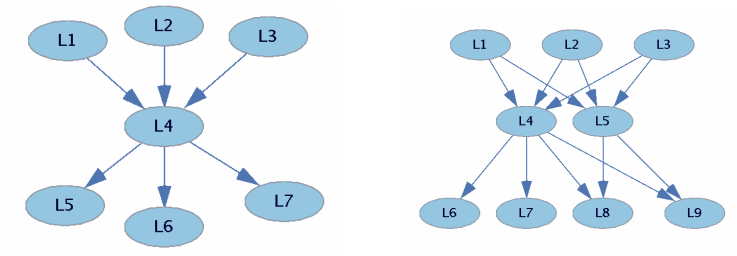

Fig. 2. a. Structure 3x1x3 and b. structure 3x2x4.

For a model, 20 different Dataset were obtained and it is intended to observe the behavior of up to three hidden variables. A total of 120 evaluations were made for all data sets. Subsequently, the values of Log loss were grouped by sample size and number of hidden variables and thus an average was obtained. Then the average Log loss of each structure was plotted by sample size and number of variables.

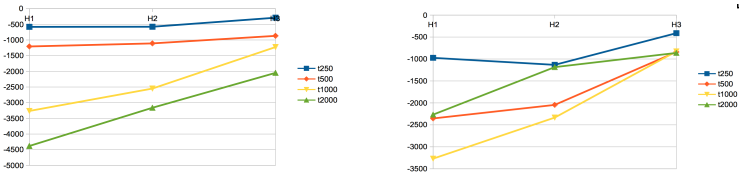

Fig. 3. Log Loss with MS-REM in 3x1x3 and 3x2x4 networks with tree hidden variables.

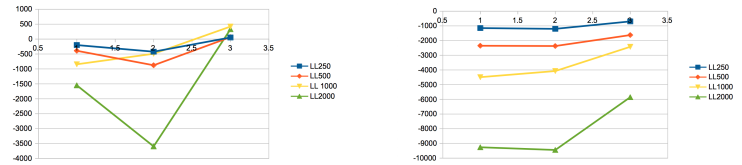

Fig. 4. Log Loss with MS-EM in 3x1x3 and 3x2x4 networks with tree hidden variables.

In these graphs (Figure 3 and Figure 4) we can see if the number of hidden variables is higher is more difficult for the algorithm to find a network with a better MDL and since it starts to find more complex networks or that do not they adequately represent the data, which causes their MDL values to be higher.

This phenomenon can be seen in the graphs in H3, the Log Loss values are closer to zero.

4.2 Performance Statistics

To test the functioning of MS-REM, it was given to different databases that are described in Table 1, all of them belonging to the UCI database, where the class variable is known, this variable was eliminated and 30 independent executions to analyze their results, MS-REM is a random algorithm and it is necessary to perform statistical tests to study its behavior [1].

Table 1. Features of the data sets. Where C represents a given class.

	Iris	B. Cancer	Car	User	Wine
Num. Classes	3	2	4	4	3
Num. Instances	150	286	1728	431	178
Num. Variables	5	10	6	7	13
Dist. Classes	C1(0.33); C2(0.33); C3(0.33)	C1(0.70); C2(0.30)	C1(0.7); C2(0.24); C3(0.03); C4(0.03)	C1(0.12); C2(0.3); C3(0.28); C4(0.3)	C1(0.33); C2(0.39); C3(0.28)

Table 2. Statistical table of the MS-REM algorithm and MS-EM with the Iris database, in relation to MDL and number of clusters (NC).

	MS-REM		MS-EM	
Iris	**MDL**	**NC**	**MDL**	**NC**
Mean	**-693.778**	**10.44**	-735.512	9.967
Median	**-677.550**	**12**	-791.589	13
Best	**-635.196**	**3**	-639.931	3
Worst	**-784.546**	**18**	-811.760	17
SD	**43.905**	**4.183**	44.1580	3.978

From the previous experiment we can see that the MS-REM algorithm presents a better behavior with respect to the original version in databases where the number of values in classes is greater and the distribution of each class is smaller. As with Iris, Wine and User. While in databases such as Car or B. Cancer where a predominant class is found, it performs worse than the original algorithm. Also the size of the database influences, in large databases such as Car the algorithm has a process of adjustment much greater than the original version.

Table 3. Statistical table of the MS-REM algorithm and MS-EM with the Wine database, in relation to MDL and number of clusters (NC).

	MS-REM		MS-EM	
Wine	**MDL**	**NC**	**MDL**	**NC**
Mean	**-3411.179**	**15.5**	-3556.06285	17.2
Median	**-3439.514**	**20**	-3444.62	15
Best	**-3264.215**	**7**	-3427.12	6
Worst	**-3554.198**	**24**	-3746.34	23
SD	**62.614**	**3.900**	139.929	4.209

Table 4. Statistical table of the MS-REM algorithm and MS-EM with the B. Cancer database, in relation to MDL and number of clusters (NC).

	MS-REM		MS-EM	
B. Cancer	**MDL**	**NC**	**MDL**	**NC**
Mean	-4573.242	17.53	**-4472.014**	**14.26**
Median	-4578.216	15	**-4425.009**	**14**
Best	-4365.966	16	**-4256.403**	**8**
Worst	-4958.715	23	**-4601.068**	**14**
SD	164.413	3.381	**126.774**	**2.947**

4.3 Convergence Graphs

Next, we will show the adjustment behavior of the MS-REM algorithm based on the MDL value from the first iteration until reaching convergence. The following graphs represent the average iteration of the performance statistics.

Through these graphs it can be seen how the algorithm starts with a low vale of MDL in all the databases because the algorithm has the worst panorama of clustering as the initial configuration and as it is approached through the adjustment functions to zero. Obtaining a better adjustment of the hidden variable with the observable data and relation with the observable variables through the Bayesian network that is changing at the same time as the means of the clusters until reaching its optimum.

Table 5. Statistical table of the MS-REM algorithm and MS-EM with the User database, in relation to MDL and number of clusters (NC).

	MS-REM		MS-EM	
User	**MDL**	**NC**	**MDL**	**NC**
Mean	**-948.357**	**11.67**	-968.727	11.4
Median	**-946.897**	**11**	-952.361	12
Best	**-906.993**	**7**	-913.339	8
Worst	**-1000.513**	**8**	-1239.14843	14
SD	**19.934**	**2.397**	72.895	2.955

Table 6. Statistical table of the MS-REM algorithm and MS-EM with the Car database, in relation to MDL and number of clusters (NC).

	MS-REM		MS-EM	
Car	**MDL**	**NC**	**MDL**	**NC**
Mean	-21511.723	26.3	**-21446.681**	**21.5**
Median	-20647.463	18	-21796.175	30
Best	-20647.463	18	**-19717.940**	**2**
Worst	-22751.763	34	**-22543.114**	**38**
SD	598.659	4.62	848.772	12.20

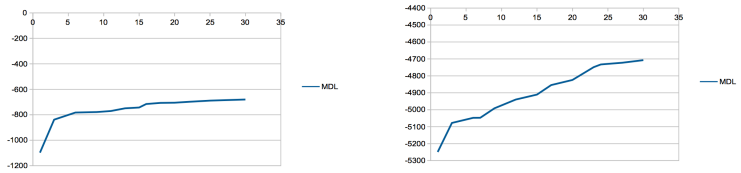

Fig. 5. Convergence graph for Iris and Breast cancer Data sets.

5 Conclusions and Discussion

In the construction of the MS-REM algorithm was presented, different tests were performed to see how to compare the functionality of the original algorithm with the one proposed in this document and see in this way if the modifications of this algorithm are better or didn't show a significant change. In the first place, a test was carried out proposed by the literature where the algorithm was executed with synthetic databases and the distribution of the data was known, we analyzed the loss of information recorded by the two algorithms. In the second place, tests the algorithms with databases of the UCI repository and the clustering process performance was analyzed of the two algorithms.

The MS-REM algorithm initialize in the worst of the cases, this allows its adjustment process to accommodate small clusters, thus causing class divisions to exist in categorical databases. However, what the author proposes is that there may be an infinite number of hidden variables within a database and that the algorithm is only able to find the "most important" at a given a initial

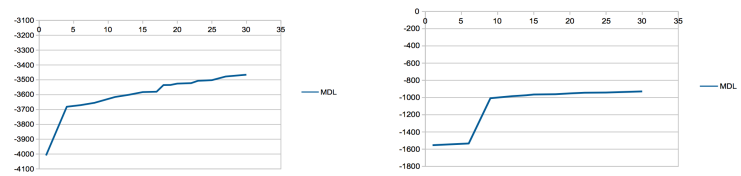

Fig. 6. Convergence graph for Wine and User knowledge modeling Data sets.

configuration and that is why the Bayesian network is changing according to the behavior of the algorithm [7].

Something that shows the literature about the hidden variables is one of its virtues is the fact that it helps us to explain a phenomenon represented in the data [8]. However, as the author mentioned in the original proposal of the algorithm, the stochastic factor within the initial configuration of EM, means that we only find the best variable for a given initial configuration, causing the study of these variables to be complex, because it is difficult to determine that the variable obtained in a certain execution is the most relevant in the database [7].

However, despite this drawback, it is an interesting proposal given that the variables found by the algorithm are obtained entirely from the data and, beyond parameter configuration of the algorithm, the user has no interaction to modify the behavior of the algorithm, thus providing interesting solutions that could explain a phenomenon, offer an alternative of how the data is being categorized or even obtain the values of a variable of great impact on the database.

References

1. Asuncion, A., Newman, D.: Uci machine learning repository (2007)
2. Bouckaert, R.R.: Probabilistic network construction using the minimum description length principle. In: European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty. pp. 41–48. Springer (1993)
3. Cruz-Ramírez, N., Acosta-Mesa, H.G., Mezura-Montes, E., Guerra-Hernández, A., de Jesús Hoyos-Rivera, G., Barrientos-Martínez, R.E., Gutiérrez-Fragoso, K., Nava-Fernández, L.A., González-Gaspar, P., Novoa-del Toro, E.M., et al.: How good is crude mdl for solving the bias-variance dilemma? an empirical investigation based on bayesian networks. PloS one 9(3), e92866 (2014)
4. Friedman, J., Hastie, T., Tibshirani, R.: The elements of statistical learning, vol. 1. Springer series in statistics Springer, Berlin (2001)
5. Friedman, N.: The bayesian structural em algorithm. In: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. pp. 129–138. Morgan Kaufmann Publishers Inc. (1998)
6. Friedman, N., Goldszmidt, M.: Learning Bayesian networks from data. Morgan Kaufmann (1999)
7. Friedman, N., et al.: Learning belief networks in the presence of missing values and hidden variables. In: ICML. vol. 97, pp. 125–133 (1997)
8. Heckerman, D.: A tutorial on learning with bayesian networks. In: Learning in graphical models, pp. 301–354. Springer (1998)
9. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier (2014)
10. Wackerly, D., Mendenhall, W., Scheaffer, R.L.: Mathematical statistics with applications. Nelson Education (2007)
11. Yang, M.S., Lai, C.Y., Lin, C.Y.: A robust em clustering algorithm for gaussian mixture models. Pattern Recognition 45(11), 3950–3961 (2012)