



A simulation experiment on bootstrap inference for Self-Organizing Maps

José Manuel Mira, María Muñoz-Fernández de Córdoba,
Gabriel Valverde and Beatriz González-Pérez

EasyChair preprints are intended for rapid
dissemination of research results and are
integrated with the rest of EasyChair.

July 1, 2021

A simulation experiment on bootstrap inference for Self-Organizing Maps

José Manuel Mira¹, María Muñoz-Fdez de Córdoba², Gabriel Valverde³, Beatriz González-Pérez⁴

¹Dpto. de Ingeniería de Organización, Administración de Empresas y Estadística –
^{1,2}ETSI Industriales – Universidad Politécnica de Madrid

³Department of Statistics and Operations Research – Complutense University of Madrid

⁴Interdisciplinary Mathematics Institute (IMI) – Complutense University of Madrid

josemanuel.mira@upm.es, maria.munozf@alumnos.upm.es,

^{3,4}{[gvalverd](mailto:gvalverd@ucm.es),[beatrizg](mailto:beatrizg@ucm.es)}@ucm.es

Summary. A stochastic simulation study is carried out to learn about quantification of uncertainties for Self-Organizing Maps (SOM). A mixture of Gaussian distributions is assumed as data generating process and the Monte Carlo generated samples are transformed according to the Kohonen (SOM) algorithm. Additionally, the original data matrix is resampled for a bootstrap quantification of parameter estimation uncertainties.

Key words. Stochastic Simulation, Self-Organizing Maps, Uncertainty, Bootstrap.

1. Introduction

The purpose of this paper is to develop a stochastic model for one-layer Self Organizing Maps (SOM), within a frequentist bootstrap framework and to quantify the associated uncertainties. Simulated data were used to illustrate the methodology and learn about its performance.

Kohonen (1980) developed SOM (Self Organizing Maps) to model structures such as cortical layers in the brain as (typically) two or in some cases three dimensional maps. From the statistical point of view, SOM is a clustering technique providing a reduction of dimensionality to typically two or three dimensions, thus producing a “visual” clustering.

Initially SOM need not be formulated in a stochastic framework. The first and “deterministic” reference to SOM is the seminal paper by Kohonen (1980), leaning on Von der Malsburg (1973) and Willshaw and von Der Malsburg’s (1976) work on competitive learning. The theoretical background is presented in Kohonen (1997). Two relevant reviews of SOM are Yin (2008), and Van Hulle (2007). Haykin (2009) provides a sound introductory reference, noting that SOM transfers probabilities as proportional to the cubic roots of the original-space ones.

However, several papers have developed a stochastic framework for SOM, the first are Luttrell (1994), Yin and Alison (1997), Lampinen and Kostianen (2001) and Lampinen and Vetari (2001); Guao et al (2013) is a more recent article. We consider that more work is necessary to enhance the stochastic formulation. Deep (more than one layer) SOM has been approached from a practical point of view in Solokowska et al (2013) and Liu et al (2015).

In this paper we use a stochastic approach in our methodological development.

1.1 Description of SOM

SOM is a neural clustering technique which assigns observations in an original p -dimensional space to integer-valued nodes in a (typically) two-dimensional map, in such way that the distances between the corresponding points in the map reflect those in the original space as much as possible. This approach is similar to multidimensional scaling (MDS), but in the latter the points in the reduced space are real numbers instead of (typically) pairs of integers. An interesting comparison between SOM and MDS is given by Trosset (2006).

SOM is also different from the traditional k -means clustering technique in that the clusters are nodes in a smaller dimensional space (than the original one), and preserving distances as much as possible, in k -means there is no dimension reduction, the cluster centroids are points in the original space, although SOM nodes have associated weights in the original space as well as the node's pairs of integers map coordinates. Besides, k -means is competitive learning and has a natural extension within the stochastic framework in Gaussian mixture models, but it does not include a cooperation step.

The stochastic model would have to coincide in mean with the (deterministic) Kohonen algorithm. Guo et al (2013) propose a probabilistic model for the weights in the SOM by means of putting probabilities on the weight distribution parameters (mean vectors and covariance matrices in a Gaussian framework), but this is not fully Bayesian since there are no priors on either these parameters or the probabilities of the p -original-space points belonging to the nodes.

A probabilistic approach would have to start with the assumption of a data generating process -dgp- also called the stochastic model, in the original space. This is a key assumption of the methodology.

In this paper, we develop a quantification of uncertainties framework for SOM within the frequentist approach, by means of a mixture of Gaussians and the bootstrap technique. The methodology provides a means of quantifying the two sources of uncertainty inherent to the problem: the first elicits the uncertainty in the data generating process and the second in parameter estimation. The procedure will require dividing the original Monte Carlo generated sample of the mixture of Gaussians into a set of subsamples, as will be explained in detail below.

The rest of this paper is organized as follows: in section 2, we review the Kohonen algorithm. In section 3 we describe the stochastic modelling and the procedure for quantification of uncertainty. In section 4 we present the results of the application of the procedure to a mixture of 6 3-dimensional Gaussian distributions, for illustration of the methodology.

2. The original (deterministic) SOM -Kohonen algorithm

Given a sample y_{i1}, \dots, y_{in} of n observations in a p -dimensional space, the Kohonen algorithm assigns each sample element to each of one of the $I * J$ nodes in the map, and produces $I * J$ so-called weights, which are representations of the nodes in the original p -dimensional space, in such way that the closer the nodes are in the map, the more similar their weights will be.

The steps of the SOM algorithm are as follows:

1. Generate initial values for the p -dimensional node weights $y_{10}, \dots, y_{I*J,0}$, where I, J are the dimensions of the map.
2. For the first observation y_1 , calculate the Euclidean distance from y_1 to the initial node weights; assign y_1 to the node with the nearest weight, which we call the winning node G .
3. Update all weight nodes through the expression:

$$y_{i1} = y_{i0} + \eta \Lambda(i, G) * (y_{i0} - y_{iG}), i = 1, \dots, I * J, \quad (1)$$

where:

y_G is the weight of the winning node G .

η is the learning rate.

$$\Lambda(i, G) = \exp(-d(i, G)/(2\sigma_\Lambda^2(t))), \quad (2)$$

$$\text{with } \sigma_\Lambda(t) = \sigma_{\Lambda_0} \exp(-\eta_0 t/n) \quad (3)$$

$$d(i, G) = \sqrt{(x_{i1} - x_{G1})^2 + (x_{i2} - x_{G2})^2} \quad (4)$$

where the x_j are the integer indices of the map

4. Repeat iteratively 2) and 3) for all the remaining observations y_2, \dots, y_n of the sample.

The parameters of the Kohonen algorithm are thus σ_{Λ_0} and η_0 . σ_{Λ_0} is a constant to calibrate the neighborhood structure for the update process. η_0 is the rate at which the updating process slows down as the algorithm approaches convergence.

3. The stochastic model

3.1 Introduction

It is assumed that there exists a data generating process or probability distribution in the original dimension of the data (p). This distribution is transformed, by means of the Kohonen algorithm, into a stochastic process, the index of which is the iterations of the algorithm, in such way that we formulate the estimation of a stochastic model in terms of the (estimation of) limit distribution to which the process converges. For given values of the parameters η and σ_{Λ_0} of the Kohonen algorithm, the problem is thus just of transformation of variables, but the above-mentioned parameters should also be estimated within an inferential process. The marginal distributions of the node weights along the stochastic process are a mixture distribution with as many components as map nodes. Note that the mixture distribution is not on the original variables, but on the Kohonen algorithm-transformed ones.

We could then follow two procedures for estimation of the final distribution of the stochastic process: 1) Estimate a distribution in the original space and then draw samples from it, applying the Kohonen algorithm to them in such a way that we obtain an -as large as desired- sample of the stochastic process. 2) Divide the original sample in N several independent subsamples and apply the Kohonen algorithm to each of these N subsamples in such way that we also obtain a sample from the stochastic process. In both cases, we use the final iteration of the sample to estimate the distribution of the (last iteration) weights which would be taken as from the distribution to which the SOM converges. Both 1) and 2) are conditional on given values of the Kohonen algorithm parameters. Here we follow the latter solution by applying in turn a frequentist procedure for quantification of parameter estimation uncertainty, based on bootstrapping.

3.2 Estimating the two sources of uncertainty

3.2.1 Estimating the data generating process

Here we wish to estimate the parameters of the mixture of Gaussians corresponding to the node weights of the last iteration, as well as the optimal values of the Kohonen algorithm calibration parameters σ_{Λ_0} and η_0 .

Our starting point is the original sample of size 90000 (y_1, \dots, y_n), where each vector y_i is 3-dimensional. If, with the purpose above, we applied the SOM algorithm to the full original sample, we would just obtain a single value for the “last iteration node weights”, which would just be a sample of size one, obviously not what we need to estimate a distribution. Bearing this in mind, we have divided the sample of size 90000 in 300 subsamples of size $90000/300=300$. We then apply the Kohonen algorithm to each of these 300 subsamples and keep the 300 last iteration node weights. These values constitute our desired sample for “last iteration node weight distribution” estimation.

Let us denote this sample with $y^* = (y_1^*, \dots, y_{n_s}^*)$, where $n=300$. We then apply maximum likelihood to y^* to estimate the distribution by assuming a Gaussian mixture distribution as well (i.e., the same kind of statistical model we used to generate our original data). We will thus obtain 6 vectors of 3-component vector means, 6 variance-covariance matrices and a vector of six mixture component probabilities. Optimal values for the Kohonen algorithm calibration parameters σ_{Λ_0} and η_0 will also be estimated. If we assume that the variance-covariance matrix is the same for all mixture components, we only need to estimate a single variance-covariance matrix instead of 6.

The likelihood function for the mixture model will be expressed in terms of y^* .

$$l = \prod_{i=1}^{n_s} \sum_{k=1}^6 \theta_k \left(\frac{1}{|\Sigma_k|} \right) \exp \left(-(y_i^* - \mu_k)^T \Sigma_k^{-1} (y_i^* - \mu_k) \right) \quad (5)$$

For point estimation, the likelihood function above is maximized with respect to all the parameters, *i. e.* $(\eta_0, \sigma_{\Lambda_0}, \theta_i, \mu_i, \Sigma_i)$.

3.2.2 Parameter estimation uncertainty

In addition to stochastic model uncertainty, there is parameter estimation uncertainty which is quantified by means of the bootstrap technique.: using the original data matrix y as starting point, 50 new data matrices are obtained with the same dimension as y , by (bootstrap) resampling, *i.e.* selecting from the rows with replacement. We then apply to each of these 50 matrices the same procedure as was done for the original data matrix in section 3.2.1. The result will be a matrix of 50 rows (one for each resampling) and as many columns as estimated parameters. Finally, this matrix is used for parameter estimation uncertainty.

4 Application to simulated data

4.1 The experimental design

A mixture of 6 tri-variate Gaussian distributions has been simulated, where the means are given in table 1. For each of the 6 mixture components, the 3 variables are independent, and the variances are equal to 2 for the 3 variables of all the 6 components.

Table 1. Mixture means for simulation experiment

y_1	y_2	y_3
10	2	3
10	7	8
12	12	13
12	8	12
13	19	15
30	30	25

Table 2. Mixture component probabilities

Component	Probability
1	0.16
2	0.16
3	0.16
4	0.20
5	0.15
6	0.17

4.2 Results

We obtain a matrix of values for the parameters. The number of rows is the number of bootstrap resampling's. The number of columns is the number of parameters estimated.

The list of parameters is: the $k \times p$ mixture means $\mu_{ij}, i = 1..K, j = 1..p$, the $k \times p(p + 1)/2$ elements of the k covariance matrices $\Sigma_i, i = 1..K$, the k mixture weights $\theta_i, i = 1..K$ and finally, the Kohonen algorithm parameters $\eta_0, \sigma_{\Lambda_0}$.

4.3 Summary of results

As mentioned above, the results of the bootstrap procedure are a matrix of estimates of the parameters, with as many rows as bootstrap resampling's and as many columns as parameters. A descriptive statistic of the most relevant (for sparcity reasons, we no not present the covariances) for marginal distributions of this sample is shown as follows.

Table 3. Results of the estimation of node weight means.

Name	n	Mean	Sd	Median	Mad	Min	Max	Skew	Kurtois
X ₁₁	50	10.00	0.04	10.00	0.03	9.86	10.15	0.16	5.12
X ₁₂	50	2.22	0.93	2.04	0.03	1.97	6.95	4.56	19.26
X ₁₃	50	3.20	0.88	3.03	0.04	2.93	8.26	4.83	22.69
X ₂₁	50	10.32	0.94	10.16	0.72	8.82	13.95	1.32	2.49
X ₂₂	50	7.11	1.44	7.01	0.17	2.00	11.30	-0.63	6.25
X ₂₃	50	8.76	2.11	8.23	0.81	2.96	13.74	0.08	1.07
X ₃₁	50	11.18	1.17	11.27	0.70	8.90	13.76	0.30	-0.33
X ₃₂	50	8.07	1.50	7.30	0.63	6.68	12.59	1.07	0.00
X ₃₃	50	10.61	1.99	11.04	1.95	7.13	13.92	-0.38	-1.16
X ₄₁	50	12.11	1.23	11.37	0.64	10.59	13.98	0.43	-1.63
X ₄₂	50	10.24	1.03	10.35	0.69	7.17	12.47	-1.55	2.94
X ₄₃	50	13.31	0.95	13.36	0.49	10.64	15.57	-0.35	1.68
X ₅₁	50	12.81	1.25	13.38	0.67	9.75	14.00	-0.81	-0.98
X ₅₂	50	10.22	0.62	10.20	0.25	6.83	11.10	-3.04	15.48
X ₅₃	50	13.77	1.17	13.81	0.63	7.62	15.53	-2.57	13.02
X ₆₁	50	29.97	0.03	29.97	0.02	29.91	30.03	-0.25	-0.29
X ₆₂	50	30.01	0.03	30.01	0.03	29.92	30.09	-0.25	0.13
X ₆₃	50	24.98	0.03	24.97	0.03	24.90	25.06	0.30	-0.42

Table 4. Results for the node weights standard deviations

Name	n	Mean	Sd	Median	Mad	Min	Max	Skew	Kurtois
sdX ₁₁	50	2.72	0.09	2.71	0.09	2.55	3.08	1.15	2.77
sdX ₁₂	50	2.86	0.17	2.82	0.10	2.62	3.77	3.12	14.88
sdX ₁₃	50	2.80	0.18	2.80	0.10	1.76	3.05	-3.74	20.01
sdX ₂₁	50	2.45	0.49	2.64	0.21	1.09	3.02	-1.50	1.07
sdX ₂₂	50	2.85	0.77	2.82	0.63	1.80	6.27	2.17	6.58
sdX ₂₃	50	3.37	0.83	3.18	0.66	2.39	6.20	1.49	2.36
sdX ₃₁	50	2.06	0.68	2.23	0.84	0.96	3.62	-0.01	-1.21
sdX ₃₂	50	3.54	1.69	2.98	1.22	1.59	6.91	0.89	-0.79
sdX ₃₃	50	3.62	1.04	3.53	1.21	2.17	6.15	0.66	-0.38
sdX ₄₁	50	1.51	0.58	1.28	0.22	0.98	2.86	1.35	0.20
sdX ₄₂	50	4.31	1.07	4.22	0.49	1.93	6.54	0.00	0.30
sdX ₄₃	50	3.73	0.64	3.96	0.42	2.14	4.66	-1.20	0.58
sdX ₅₁	50	1.54	0.63	1.24	0.22	0.96	3.00	1.28	0.00
sdX ₅₂	50	4.31	0.78	4.31	0.62	2.92	6.72	0.97	1.09
sdX ₅₃	50	3.63	0.75	3.95	0.43	2.08	4.60	-0.89	-0.68
sdX ₆₁	50	2.84	0.08	2.84	0.08	2.61	3.01	-0.29	0.32
sdX ₆₂	50	2.74	0.09	2.73	0.10	2.58	2.94	0.13	-0.78
sdX ₆₃	50	2.79	0.09	2.80	0.09	2.59	2.95	-0.28	-0.70

Table 5. Results for the mixture component probabilities

Name	n	Mean	Sd	Median	Mad	Min	Max	Skew	Kurtois
Prob1	50	0.16	0.01	0.16	0.00	0.08	0.18	-5.52	34.55
Prob2	50	0.15	0.03	0.15	0.04	0.09	0.20	-0.10	-1.26
Prob3	50	0.15	0.03	0.15	0.03	0.08	0.21	-0.03	-0.60
Prob4	50	0.19	0.03	0.19	0.02	0.10	0.23	-1.09	0.36
Prob5	50	0.19	0.03	0.20	0.02	0.11	0.23	-1.02	0.06
Prob6	50	0.17	0.00	0.17	0.00	0.17	0.17	-0.15	-0.59

Table 6. Results for the Kohonen algorithm parameters

Name	n	Mean	Sd	Median	Mad	Min	Max	Skew	Kurtois
etaoptimo	50	0.04	0.01	0.03	0.02	0.02	0.07	0.51	-1.22
sigmaoptimo	50	4.38	0.03	4.24	2.03	2.21	6.91	0.10	-1.37

5. Discussion of the results

We observe that the means are very close to those of the original mixture distribution, with low variability. The estimates for the variances are significantly less close to those of the mixture and with also significantly larger variability. The mixture component probabilities are very close to the original ones, and so the Kohonen algorithm parameters, although the estimation procedure was more directed to trying values in the range of the values estimated for the original (non-bootstrapped) data matrix.

References

- [1] Arsuaga Uriarte, E., and Del Moral, F., "Topology Preservation in SOM", World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:2, No:9, 2008, 3192-3195
- [2] Dahl, D., (2009), "Modal clustering in a class of product partition models", Bayesian Statistics, 4, 2, 243-264
- [3] Durbin, R., Szeliski, R., y Yuille, A., (1989), "An analysis of the Elastic Net approach to the travelling salesman problem", Neural Computation, 1, 348-358
- [4] Fytalis, N., and Rizzo, D.M. "Stream classification studies using multiple assessment data", Water resources research vol 49, 7747--7762, doi:10.1002/2012WR013422, 2013
- [5] Guo, X., Wang, H., and Glass, D., (2013), "Bayesian Self Organizing map for data classification and Clustering", International Journal of Wavelets, Multiresolution and Information Processing, Vol. 11, No. 5 (2013) 1350037 (12 pages)
- [6] Haykin, S., (2009), "Neural networks and learning machines". Pearson Education,
- [7] Kohonen, T., (1990), "The self organizing map", Proceedings of the IEEE, vol. 78, num 9, pp 1464-1480
- [8] Lampinen, J., and Kostiainen, T., "Generative Probability Density Model in the Self-Organizing Map"
- [9] Lampinen, J. and Vehtar, A., (2001), "iBayesian approach for neural networks: review and case studies". Neural Networks 14 (2001) 257-274
- [10] Luttrell, S.P., (1994), "A Bayesian Analysis of Self-Organising Maps", Neural Computation, 1994, vol. 6, pp.767-794.
- [11] Mulier, F. y Cherkassy, V. (1995), "Statistical analysis of self-organization, Neural Networks, vol 8, num. 5, 717-727
- [12] Mulier, F., y Cherkassy, V., (1995), " Self organizing as an iterative kernel smoothing problem"
- [13] Lau, J., y Green, P (2007), "Bayesian Model-Based Clustering procedures".

- [15] Lavine, M., and West, M., (1992), "A Bayesian Method for Classification and Discrimination". The Canadian Journal of Statistics, vol. 20, No.,4, pp-451-461.
- [16] Nowlan, S., "Maximum likelihood competitive learning"
- [17] Ormonitei, D. y Tresp, V., "Improved Gaussian Mixture density estimates using Bayesian penalty terms and network averaging"
- [18] Ruz, G.A., Truong Pham, D., (2012), "NBSOM: The naive Bayes self-organizing map", , Neural Comput & Applic (2012) 21:1319--1330
- [19] DOI 10.1007/s00521-011-0567-9
- [20] Shiwani, S., and Ganga, G., (2014), "Comparison between Unsupervised Feature Learning Methods Using an Auto-Encoder and Self-Organizing Map Algorithm for Sign Language Recognition", Volume 4, Issue 12, December 2014 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering
- [21] Sokolovska, N., Thanh Hai, N., Clement, K., and Zucke, J.D., "Deep Self-Organising Maps for Efficient Heterogeneous Biomedical Signatures Extraction"
- [22] Theodoridis, S., y Koutroubas, K., (2009), "Pattern Recognition. Academic Press
- [23] Theodoridis, J., (2015), "Machine Learning: A Bayesian and Optimization Perspective". Academic Press.
- [24] Trosset, M., (2006), "Representing Clusters: K-Means Clustering, Self-Organizing Maps, and Multidimensional Scaling". Department of Mathematics, College of William & Mary, P.O. Box 8795, Williamsburg, VA.
- [25] Utsugi, A., (1996), "Hyperparameter selection for self-organizing maps", Em parecido a SOM batch learning
- [26] Van Hulle, M., Self-Organizing Maps, chapter 1 de un libro
- [27] von der Malsburg, C., (1973), "Self-Organization of Orientation Sensitive Cells in the Striate Cortex", Kybernetik 14, 85--100
- [28] Wang, L., y Dunson, D., (2011), "Fast Bayesian inference in dirichlet process mixture models", , Journal of computational and graphical statistics, vol 20, num 1, 196-216 incremental learning

- [29] Wei, W. and Khan, A., Shukla, N, "MaxMin-SOMO: A SOM Optimization Algorithm for Simultaneously Finding Maximum and Minimum of a Function", , parece technical report
- [30] Yin, H, and Allison, N.M., (1997), bayesian learning of self organising maps, electronics Letters, vol33, num 4, pp 304-305
- [31] Yin, H, and Allison, N.M., (2001), Bayesian self-organising map for Gaussian mixtures, IEE proceedings visual image signal processing, vol 148, num 4, 234-240
- [32] Yin, H., (2008), "The Self-Organizing Maps: Background, Theories, Extensions and Applications", , Studies in Computational Intelligence (SCI) 115, 715--762
- [33] Yin, H. and Allison, N., (1997), "Comparison of a Bayesian SOM with the EM algorithm for gaussian mixtures", probabilista no bayesiano de incremental SOM pero sin dar modelo probabilista de verdad coherente con las ecs de actualizaci\{o}n
- [34] Yin, H., (2008), "The Self-Organizing Maps: Background, Theories, Extensions and Applications", , Studies in Computational Intelligence (SCI) 115, 715--762
- [35] Yuille, A., (1990), "Generalized Deformable models, Statistical physics and matching problems", Neural Computation, 2, 1-24
- [36] Yuille , A., Solortz, P., y Utans, J., (1994), Statisticall physics, mixture of distributions and the EM algorithm