SOM+ : a monitored SOM and its application to Bioinformatics

Susana Vegas-Azcárate
Statistics and Decision Sciences Group
University Rey Juan Carlos
28933 Móstoles, Spain
susana.vegas@urjc.es

Jorge Muruzábal in memoriam
Statistics and Decision Sciences Group
University Rey Juan Carlos
28933 Móstoles, Spain

Abstract - *Self-organizing maps have become standard tools for frequently encountered data-analytic tasks such as visualization, clustering and classification. Unfortunately, however, a complete SOM training methodology is not firmly established yet. When using the standard SOM training method, it is well-known that an appropriate choice of the final adaptation radius is crucial for obtaining topology-preserving maps. To avoid phase transitions, the neighborhood range at the end of training must create a tradeoff between the approximation accuracy of weight vector distribution and the stability ordering. There exist several metrics for 'monitoring' the training process, from which optimization schemes have been put forward for the neighborhood cooling scheme. However, the usual topology-preservation metrics are not very sensitive to small topological defects, and, furthermore, they require a lot of computational effort as a monitoring tool. Here, we propose the SOM+ training algorithm, the first methodology that estimates the decrementing schedules for the neighborhood range function during the training automatically, monitoring to lower the risk of phase transitions. The corresponding topological map so obtained with SOM+ should be very useful in organizing large protein or DNA databases and for rapidly classifying new sequences, since SOM+ leads automatically to the map that explains in the best way the real population.*

Key words - Bioinformatics, Topology-preserving Self-Organizing Maps

SOM+ : a monitored SOM and its application to Bioinformatics

S. Vegas-Azcárate and J. Muruzábal in memoriam

Statistics and Decision Sciences Group
University Rey Juan Carlos, 28933 Móstoles, Spain
susana.vegas@urjc.es

ABSTRACT

Self-organizing maps have become standard tools for frequently encountered data-analytic tasks such as visualization, clustering and classification. Unfortunately, however, a complete SOM training methodology is not firmly established yet. When using the standard SOM training method, it is well-known that an appropriate choice of the final adaptation radius is crucial for obtaining topology-preserving maps. To avoid phase transitions, the neighborhood range at the end of training must create a tradeoff between the approximation accuracy of weight vector distribution and the stability ordering. There exist several metrics for 'monitoring' the training process, from which optimization schemes have been put forward for the neighborhood cooling scheme. However, the usual topology-preservation metrics are not very sensitive to small topological defects, and, furthermore, they require a lot of computational effort as a monitoring tool. Here, we propose the SOM+ training algorithm, the first methodology that estimates the decrementing schedules for the neighborhood range function during the training automatically, monitoring to lower the risk of phase transitions. The corresponding topological map so obtained with SOM+ should be very useful in organizing large protein or DNA databases and for rapidly classifying new sequences, since SOM+ leads automatically to the map that explains in the best way the real population.

MOTIVATION

The visualization of large protein and DNA databases in a compact way may give insights into the data, leading to the development of new ideas and theories. Since the number of known DNA and proteins sequences is growing exponentially as a result of Genome projects [1]–[3], the management of the resulting databases is of central interest in modern Bioinformatics analysis. Many powerful algorithms for comparing two [4,5] or more proteins [6]–[9] have been developed. Although these methods are sensible, they are extremely time consuming. Faster but less precise algorithms for searching homologies have been proposed [10]–[13]. In this way, a variety of neural networks have been used to organize protein sequences into clusters or families according to their sequence homologies. Since the number and composition of the families are not known, the use of unsupervised learning algorithms, such as the SOM [14] type algorithms, seems indeed very appropriate. The ordered grid it produces can be used as a visualization surface and a number of techniques have been proposed to visualize cluster structures of data for various purposes [14]–[22]. However, only a SOM free of topological defects provides a successful tool to visualize clusters [23]–[29].

It is essential to have maps with good topological order, since contiguous clusters in the input distribution could otherwise be considered as separate clusters in the visualization of the topological map. During training, the map changes from a large neighborhood range, in which the mapping is general, to a small one, in which the mapping becomes specific. A crucial factor for a successfully trained topographic map is the 'cooling' scheme, which determines the rate at which the neighborhood range decreases over time [14,16,30]–[33].

To avoid phase transition, the neighborhood range at the end of the training process must be large enough, thus developing a 'smooth' weight distribution [34], that is, the neighborhood range must create a tradeoff between the approximation accuracy of weight vector distribution and the stability ordering.

Although it is commonly suggested that the neighborhood range should decrease sufficiently slow to a final value, which is sufficiently high, the open questions are how to determine the appropriate rate at which the neighborhood function range decreases, and also the final value of the neighborhood range. Intuition and common sense may provide a few rules of thumb, but it would be desirable to have principled and iterative methods for making these choices. Thus, measures are required to estimate the degree of topology preservation of the map. These would then allow, in principle, for the optimization of the training process.

Topology preservation measures, such as the one introduced by [35] and further studied by [36] among others, do not depend on local stretching of the lattice but on large-scale violations of the topographic ordering, and, due to their computational cost, they can not be used for monitoring the degree of topology-preservation achieved during learning. Furthermore, only for cases in which input and output space have the same dimension, the global order of the lattice can be uniquely characterized.

In the case of the standard SOM,

$$\Delta w_i = \eta \cdot \Lambda(\sigma_\Lambda(t)) \cdot (v_n - w_i),$$

where $\{v_n, w_i\} \in \mathbb{R}^d$ denote the input data and weight vectors, respectively, and Λ the neighborhood function, typically chosen with a Gaussian shape and a monotonically decreasing range $\sigma_\Lambda(t)$, several heuristics have been suggested to guide the choice of the neighborhood range. Interesting criteria for the evaluation of the degree of ordering have been developed by [37]–[41]. However, is noted in [40,33] that the computation of these measures may be computationally demanding in general. Still, [41] point out that the main regularization issue is far from settled: "the map may start to overfit as soon as the neighborhood is reduced to any practical level, indicating that some other forms of regularization may also be needed".

An alternative heuristic is proposed and tested in [42]–[44] for the monitoring of topographic maps. The proposed monitoring criterion— called UDL criterion, since is based on the Uniformity of the DataLoad vector— is a new systematical way to monitor the degree of topology-preservation during learning by adjusting the decreasing rate of the neighborhood

range. UDL scheme is able to obtain topology-preserved maps from all topographic map formation algorithms [33]. Moreover, it was shown that this method is optimal with respect to density estimation [45].

SOM+ TRAINING ALGORITHM

During the development of a topographic map, two undesired phase transitions can occur, one due to a topological mismatch between lattice and data manifold, the other due to the transition from principal curve mapping to overfitting [30]–[32,14,16]. These phase transitions can happen, respectively, when the neighborhood function range decreases too rapidly due to which the map has insufficient time to unfold itself to the principal manifold, and when the final value of the neighborhood range is too small due to which the neighborhood influences become too small and the mapping becomes too specific. Thus, an ideal cooling scheme should be *slow enough*, such that the map can unfold properly, and the final value of the neighborhood range should be *large enough*, such that overtraining is avoided. In this context, the 'monitoring' process can be defined as iteratively refining the cooling scheme, such that it is slow enough and the final value of the neighborhood function is large enough.

The predictive log-likelihood is based on test data extracted from the same source but not used for training— a kind of cross-likelihood. When training the map, the second type of phase transition occurs when the neurons begin to learn more about the concrete train set than about the underlying density function, adding noise to the final estimated distribution. The aim of the stopping criterion is to find the map with maximum predictive likelihood in which overtraining has not appeared yet and the resulting neurons' distribution provides the map with the highest likelihood. This map will be the one that explains in the best way the real population, and not just the sample. Moreover, at the cycle which corresponds to the maximum predictive likelihood, the map is completely disentangled— at the moment of maximum predictive likelihood, topographic order is maintained [42]. From this moment on, overtraining problems arise, since the learning process stresses local regions instead of global relationships, and this is reflected on both the loss of organization and a lower predictive likelihood value.

Consequently, the neighborhood value for which the maximum predictive likelihood function is reached is the one with which the training process should be ended in order to avoid the phase transition due to overtraining. This value is the optimal one, provided that an infinitely slow training, that begins with an unfolded map, is developed.

Unfortunately, the predictive log-likelihood function presents some difficulties when used as a monitoring tool during the learning process [42]. First, a representative test set is not always easy to find in real-world examples. Second, the estimated density function at each test vector has to be computed at every cycle of training, leading to a computationally expensive monitoring process.

In Ref. [42] the 'uniform dataload' vector (UDL) is proposed as a computationally cheap alternative to the log-likelihood (early reports have been made in [44]). Originally, UDL criterion looked for the moment at which the speed of decrease of the dataload standard deviation function is nearly zero. A new heuristic is developed in [42] for determining the point at which the map is most likely to correspond to that found using the log-likelihood criterion.

Dataloads— the number of data vectors projecting onto each trained unit— are the natural estimate of the probability of activation given new data generated by the sampling distribution [46]. In the truly equiprobabilistic case— whereby all neurons have an equal probability to be maximally excited [16]— the dataload vector should be distributed as a multinomial with equal component probabilities, since the pointer density in the trained equiprobabilistic map serves as an estimate of the density underlying the data. From an information-theoretic point of view, equiprobabilistic maps transfer the maximum amount of information available about input distribution, leading to a 'faithful' representation [47] of the sampling distribution. This way, in the truly equiprobabilistic case each neuron would cover about the same proportion of the data, so the dataload vector would follow asymptotically a joint multivariate Gaussian distribution with correlations getting weaker with increasing sample size [44].

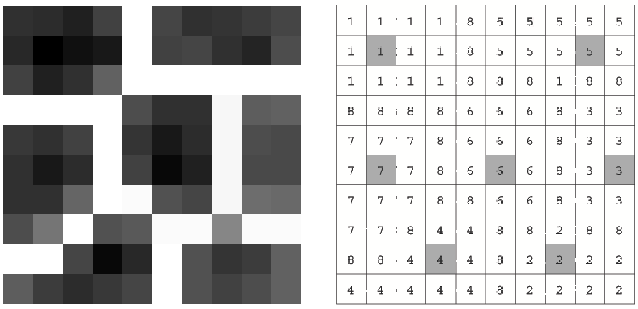
It appears that the stochastic Gaussian behavior in the equiprobabilistic case can be approximately detected when it is first reached [44]. Hence, the associated UDL stopping rule could be stated as follows: quit as soon as the trained map shows the first signs of having reached the reference Gaussian DL distribution— a moment referred to as the UDL stage. In other words, quit as soon as the dataload standard deviation function reaches its stability. Furthermore, minor gains in quantization error brought about by training beyond the UDL stage seem to enforce the loss of useful organization, implying fuzzier displays for analysis. Indeed, the UDL stage also signals approximately the beginning of the fine-tuning phase in quantization error [44].

The previous result can now be incorporated into a monitoring scheme similar to that presented in [16]. Each monitoring run consists of the training of the map following a given cooling scheme. After each monitoring run, this cooling scheme is adjusted in such a way that the probability of a phase transition is expected to be smaller— longer training and terminating training when the optimal neighborhood function range is reached. Hence, in a finite and affordable number of training cycles, the same optimal map as that achieved in an infinitely slow training, is obtained. The following neighborhood cooling scheme is used,

$$\sigma_\Lambda(t) = \sigma_\Lambda(0) \cdot \exp\left(-2 \cdot \sigma_\Lambda(0) \cdot \frac{t}{T_{udl}} \cdot \gamma_{udl}\right), \quad (1)$$

where $\sigma_\Lambda(0)$ is the initial neighborhood range, γ_{udl} the parameter that controls the slope of the cooling scheme and T_{udl} the number of cycles needed to reach the optimal neighborhood function range, in other words, where γ_{udl} and T_{udl} provide the map that has the optimal neighborhood range value.

After each monitoring iteration, the cooling scheme is adjusted such that it better approximates, in loose terminology, the relevant part of the infinitely long training process, by increasing the number of epochs to reach the optimal neigh-



a b

Fig. 1. Performance by SOM+ on the '7 clusters in 5D space' data set, (a) Density matrix; (b) Labels matrix combined with the local maxima of the Hill-Climbing procedure

borhood range determined in the current monitoring iteration. Indeed, it has been shown that using a slow enough cooling scheme, the neighborhood range value at which the dataload standard deviation function reaches its stability exists [44]. The convergence of the monitoring process can be evaluated by observing the convergence of the minimal dataload standard deviation value [42].

Specifically, while the neighborhood range function $\sigma_\Lambda(t)$ in the standard SOM is usually chosen ad hoc by the user, in SOM+ learning rule $\sigma_\Lambda(t)$ follows Eq. (1), leading automatically to the map that explains in the best way the real population. This way, SOM+ methodology estimates the decrementing schedules for the neighborhood range function during the training automatically, monitoring to lower the risk of phase transitions, and to approximate the relevant part of a infinitely slow cooling scheme.

An artificial data set concerning balanced Gaussian mixtures is presented. Data are generated in a two-step process. First, the locations of the Gaussian centroids is sampled, then each Gaussian is sampled in term. All Gaussians are spherical and have the same size, $\sum_{i=1}^{7} \frac{1}{7} N(\mathbf{c}_i^t, 0.1 \cdot \mathbf{I}_5)$, with $\mathbf{c}_i \sim N(\mathbf{0}_5, \mathbf{I}_5)$ independent and identically distributed. Hence, this data set consists of 2,000 input patterns living in a five-dimensional space and we look for 7 decision classes.

SOM+ algorithm obtains an accurate estimation to the seven clusters' locations (see Fig. 1). In order to support the assessment, we consider a minimum Euclidean labelling scheme, in which each neuron is marked with the label that most occurs within its activation region. Note that an extra label is needed for the case a neuron has no data projected into it. Moreover, when a map has been trained properly, the neuron positions are related to the density function underlying the training data. Therefore, a density estimate can be constructed from the map by positioning a Gaussian or other kernel at each neuron position, the width of which can be either fixed or variable. A gray-valued density matrix depicting the density values at the lattice nodes, is used to visualize the estimated density. In this matrix darker means larger. Then, steepest ascent Hill-Climbing [16] is developed on the network structure. This way, all neighbors are compared and the best is selected, until no further improvement is possible.

REFERENCES

[1] J. Watson, "The human genome project: Past, present and future," *Science*, vol. 248(4951), pp. 44–49, 1990.

[2] J. Maddox, "Ever-longer sequences in prospect," *Nature*, vol. 13, pp. 357–370, 1992.

[3] P. Stolorz, A. Lapedes, and Y. Xia, "Predicting protein secondary structure using neural net and statistical methods," *J. Mol. Biol.*, vol. 225, pp. 363–377, 1992.

[4] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.*, vol. 48, pp. 443–453, 1970.

[5] T. Smith and M. Waterman, "Comparison of biosequences," *Advances in Applied Mathematics*, vol. 2, pp. 482–489, 1981.

[6] M. S. Waterman, "General methods of sequence comparison," *Bull. Math. Biol.*, vol. 46, pp. 473–500, 1984.

[7] F. Corpet, "Multiple sequence alignment with hierarchical clustering," *Nucleic Acids Res*, vol. 16, pp. 10 881–10 890, 1988.

[8] D. Higgins, "CLUSTAL V: multiple alignment of DNA and protein sequences." *Methods Mol Biol.*, vol. 24, pp. 307–318, 1994.

[9] C. Mahabhashyam, M. Brudno, and S. Batzoglou, "PROBCONS: Probabilistic consistency-based multiple sequence alignment." *Genome Research*, vol. 15, pp. 330–340, 2005.

[10] W. Wilbur and D. Lipman, "Rapid similarity searches of nucleic acid and protein data banks," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 80, pp. 726–730, 1983.

[11] D. Lipman and W. Pearson, "Rapid and sensitive protein similarity searches," *Science*, vol. 227, pp. 1435–1441, 1985.

[12] S. Altschul and D. Lipman, "Protein database searches for multiple alignments," *Proc. Natl. Acad. Sci. USA*, vol. 87, pp. 5509–5513, 1990.

[13] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," *J. Mol. Biol.*, vol. 215, pp. 403–410, 1990.

[14] T. Kohonen, *Self-Organizing Maps*. Berlin: Springer-Verlag, 3rd extended ed., 2001.

[15] P. Somervuo and T. Kohonen, "Clustering and visualization of large sequence databases by mean of an extension of the self-organizing map," in *Proceedings of the Discovery Science*, S. Arikawa and S. Morishita, Eds. Berlin: Springer, 2000, pp. 76–85.

[16] M. M. Van Hulle, *Faithful representations and topographic maps: From distortion- to information-based self-organization*. New York: Wiley, 2000.

[17] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[18] E. A. Ferrán and P. Ferrara, "Topological maps of protein sequences," *Biological Cybernetics*, vol. 65, pp. 451–458, 1991.

[19] E. A. Ferrán and P. Ferrara, "Clustering proteins into families using artificial neural networks," *Cambios*, vol. 8(1), pp. 39–44, 1992.

[20] P. Arrigo, F. Giuliano, F. Scalia, A. Rapallo, and G. Damiani, "Identification of a new motif on nucleic acid sequence data using Kohonen's self-organizing map," *Comput. Appl. Biosci.*, vol. 7, pp. 353–357, 1991.

[21] V. Rose, I. Croall, and H. MacFie, "An application of unsupervised neural network methodology (Kohonen topology-preserving mapping) to QSAR analysis," *Quant. Struct. Act. Relat.*, vol. 10, pp. 6–15, 1991.

[22] T. Kohonen and P. Somervuo, "How to make large self-organizing maps for nonvectorial data," *Neural Networks*, vol. 15, pp. 945–952, 2002.

[23] S. P. Luttrell, "Code vector density in topographic mappings: Scalar case," *IEEE Trans. Neural Networks*, vol. 2(4), pp. 427–436, 1991.

[24] D. R. Dersch and P. Tavan, "Asymptotic level density in topological feature maps," *IEEE Trans. Neural Networks*, vol. 6, pp. 230–236, 1995.

[25] J. Lampinen and T. Kostiainen, "Self-organizing map as a probability density model," in *Proc. IJCNN' 2001. IEEE International Joint Conference on Neural Networks. Washington, D.C, USA.*, 2001.

[26] T. Kostiainen and J. Lampinen, "On the generative probability density model in the self-organizing map," *Neurocomputing*, vol. 48, pp. 217–228, 2002a.

[27] J. Lampinen and T. Kostiainen, "Generative probability density model in the self-organizing map," in *Self-organizing neural networks: Recent advances and applications*, U. Seiffert and L. Jain, Eds. Physica Verlag, 2002b, pp. 75–94.

[28] M. M. Van Hulle, "The formation of topographic maps that maximize the average mutual information of the output responses to noiseless input signals," *Neural Computation*, vol. 9(3), pp. 595–606, 1997a.

[29] M. M. Van Hulle, "Nonparametric density estimation and regression achieved with topographic maps maximizing the information-theoretic entropy of their outputs," *Biol. Cybern.*, vol. 77, pp. 49–61, 1997c.

[30] R. Der and M. Herrmann, "Phase transitions in self-organizing feature maps," in *Proc. ICANN'93 (Amsterdam, The Netherlands)*. New York: Springer, 1993, pp. 597–600.
[31] R. Der and M. Herrmann, "Inestabilities in self-organized feature maps with short neighborhood range," in *Proc. Eur. Symp. on Artificial Neural Networks - ESANN'94 (Brussels, Belgium)*, M. Verleysen, Ed., 1994, pp. 271–276.
[32] K. Haese and G. J. Goodhill, "Auto-SOM: Recursive parameter estimation for guidance of self-organizing feature maps," *Neural Computation*, vol. 13, pp. 595–619, 2001.
[33] S. Vegas-Azcárate, T. Gautama, and M. Van Hulle, "Topology preservation in topographic maps," *Technical Reports on Statistics and Decision Sciences, Rey Juan Carlos University*, vol. TR05/14, 2005.
[34] F. Mulier and V. Cherkassky, "Self-organization as an iterative kernel smoothing process," *Neural Computation*, vol. 7, pp. 1165–1177, 1995.
[35] H.-U. Bauer and K. Pawelzik, "Quantifying the neighborhood preservation of self-organizing feature maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 570–579, 1992.
[36] T. Villmann, R. Der, M. Herrmann, and T. Martinetz, "Topology preservation in self-organizing feature maps: Exact definition and measurement," *IEEE Trans. Neural Networks*, vol. 8(2), pp. 256–266, 1997.
[37] T. Kohonen and P. Somervuo, "Self-organizing maps of symbol strings," *Neurocomputing*, vol. 21, pp. 19–30, 1998.
[38] T. Villmann, R. Der, and T. Martinetz, "A new quantitative measure of topology preservation in Kohonen's feature maps," *In Proc. of the IEEE Int. Conf. on Neural Networks (ICNN'94)*, pp. 645–648, 1994.
[39] S. Zrehen, "Analyzing Kohonen maps with geometry," *Proc. Int. Conf. on Artificial Neural Networks (ICANN'93)*, pp. 609–612, 1993.
[40] S. Kaski and K. Lagus, "Comparing self-organizing maps," in *Proceedings of ICANN'96, International Conference on Artificial Neural Networks, Lecture Notes in Computer Science*, C. von der Malsburg and B. Sendhoff, Eds., vol. 1112. Springer, Berlin, 1996, pp. 809–814.
[41] J. Lampinen and T. Kostiainen, "Overtraining and model selection with the self-organizing map," in *IEEE, Proc. IJCNN'99, Washington, DC, USA*, 1999, pp. 1911–1915.
[42] S. Vegas-Azcárate, T. Gautama, and M. Van Hulle, "Monitoring topographic maps," *Technical Reports on Statistics and Decision Sciences, Rey Juan Carlos University*, vol. TR05/13, 2005.
[43] S. Vegas-Azcárate and J. Muruzábal, "On cluster analysis via neuron proximity in monitored self-organizing maps," in *1st Workshop on Biosignal Processing and Classification, Barcelona, Spain*, 2005.
[44] J. Muruzábal and S. Vegas-Azcárate, "On equiprobabilistic maps and plausible density estimation," in *5th Workshop On Self-Organizing Maps, Paris*, 2005.
[45] S. Vegas-Azcárate and J. Muruzábal, "Density estimation with equiprobabilistic maps," *Technical Reports on Statistics and Decision Sciences, Rey Juan Carlos University*, vol. TR06/03, 2006.
[46] E. Bodt, M. Verleysen, and M. Cottrell, "Kohonen maps versus vector quantization for data analysis," in *European Symposium on Artificial Neural Networks, ESANN'97*. Brussels: D-Facto publications, 1997, pp. 211–220.
[47] J. K. Lin, D. G. Grier, and J. D. Cowan, "Faithful representation of separable distributions," *Neural Computation*, vol. 9, pp. 1305–1320, 1997.