

LOAD PROFILING BY SELF ORGANIZATION WITH AFFINITY CONTROL STRATEGY

MIHAI GAVRILAȘ, OVIDIU IVANOV

Key words: Load profiling, Self organization, Fuzzy logic, Electricity distribution.

This paper describes a modified self-organizing algorithm which addresses the problem of consumer classification in electric distribution networks according to the shape of the load profiles. This algorithm combines the traditional growing self organizing map algorithm with fuzzy implementation and a control strategy based on affinity measures.

1. INTRODUCTION

The way consumers are represented in the general model of the network is largely responsible for the efficient planning and operation of distribution networks. It is generally accepted that during the planning stage the usage of global (e.g. peak load, time of peak load, time of losses) and statistical (e.g. simultaneity coefficients) parameters of the load profiles offers good results. However, during the operation stage, decisions like sectionalizing open-loop networks, reactive power compensation, optimal tap position for transformers in substations, or post-fault network reconfiguration can no more be made based on the above methodology.

The literature suggests using typical load profiles (TLPs) associated with simple readings in the network as models for end-use consumers. Standard procedures generate TLPs by off-line processing a great amount of load data recorded at consumers with known, simple structure. TLPs are represented using 60, 30 or 15 minutes sampling rates and are expressed in kW or p.u. [4, 5]. Self-organizing algorithms applied to load profile classification offer certain advantages, such as the simultaneous development of the classification and TLP generation processes or the avoidance of human interference during classification.

This paper proposes a new technique for consumer classification and TLPs data base generation. A fuzzy implementation of Self Organizing Maps (SOMs) is applied as a basic approach, combined with a weighting procedure to compute

“Gh. Asachi” Technical University of Iasi, Bd. D. Mangeron 53, 700050 Iasi, Romania;
mgavril@ee.tuiasi.ro

distances between metered load profile and TLPs around peak and valley load hours previously developed by the authors [3]. Then the growing SOM technique is applied to determine the optimal architecture of the SOM, using a control strategy based on the assessment of affinity measures between metered load data and TLPs.

2. PRINCIPLES OF SELF ORGANIZATION

Self-organization, sometimes called unsupervised learning, is one of the Computational Intelligence paradigms which describes a process of information organization without any external guidance or supervising action. Information organizes itself based only on local properties that describe it. There are a large variety of self-organizing algorithms, which sometimes use hybrid supervised-unsupervised techniques: K-means algorithm, Learning Vector Quantization (LVQ), Kohonen neural networks and the Adaptive Resonance Theory [6, 7].

However, the a priori knowledge of the number of classes to be separated, the choice of certain "vigilance" parameters or the sensitivity with respect to the order the patterns are presented during the learning stage are difficulties that must be taken into consideration.

The standard Kohonen learning algorithm produces a vector quantizer by repeatedly updating the prototypes of the class-units (see Box 1). After initialization of prototypes $W_i(t)$, learning rate $\eta(t)$ and neighbourhoods $N_i(t)$, all patterns are presented to the SOM network. For each pattern X_p ($p = 1 \dots P$), C distances $d_i = \|W_i(t) - X_p\|$ to the prototypes of the class-units are computed and the winning class-unit i^* is chosen. The prototypes of the class-unit i^* and the class-units inside the neighbourhood $N_{i^*}(t)$ are then updated.

One difficulty with the standard Kohonen algorithm is the need of a priori knowledge of the number of classes to be separated during the classification process (parameter C in Box 1). To overcome this problem a Growing Self Organizing Map (GSOM) algorithm may be used [1], as described in Box 2. During the growing phase, after a class-unit i^* is found, the deviation between the current pattern X_p and the prototype of the winning class-unit W_{i^*} is used as an error term to determine if new class-units must be added to the SOM network.

Although the GSOM algorithm surpasses the standard Kohonen algorithm in terms of network architecture design, both algorithms suffer of one more drawback, namely their sensitivity with respect to the order in which the learning patterns are presented to the network. This problem can be avoided if the prototypes updating procedure uses a fuzzy batch mode technique described in [2 and 3], and denoted by FSOM. In this case a fuzzy-Gaussian neighbourhood function is used to combine the learning rate and the neighbourhoods $N_i(t)$ into a single coefficient $\eta_{ip}(t)$, which is associated to the class-unit i and the pattern p

($I = 1 \dots C$; $p = 1 \dots P$). The FSOM algorithm uses real parameters m and m_0 which have to be determined using real data experiments [3].

Box 1

The Kohonen algorithm

- 1) Initialize the network: number of class-units C , prototypes W_i , learning rate $\eta(t)$, neighbourhoods $N_i(t)$, $i = 1 \dots C$, maximum number of epochs T_{max} and accuracy ε .
- 2) Present a new pattern X_p to the network.
- 3) Determine the winning class-unit i^* , the one closest to pattern X_p :

$$d_{i^*} = \min(d_i) = \min(\|W_i(t) - X_p\|) \quad \forall i = 1 \dots C$$
- 4) Update prototypes of the winning class-unit i^* and class-units inside the neighbourhood $N_{i^*}(t)$:

$$W_i(t+1) = W_i(t) + \eta(t)[X_p - W_i(t)] \quad i \in N_{i^*}(t)$$
- 5) Update the learning rate $\eta(t+1)$ and the neighbourhoods $N_i(t+1)$, $i = 1 \dots C$.
- 6) Repeat steps 2-5 until either the changes in prototypes in two consecutive epochs fall below ε , or the maximum number of epochs T_{max} is reached.

Box 2

The GSOM algorithm

- 1) Initialize the network: initial number of class-units C , prototypes W_i , learning rate $\eta(t)$, neighbourhoods $N_i(t)$, $i = 1 \dots C$, maximum number of epochs T_{max} and the growth threshold GT .
- 2) Present a new pattern X_p to the network.
- 3) Determine the winning class-unit i^* , the ONE closest to pattern X_p :

$$d_{i^*} = \min(d_i) = \min(\|W_i(t) - X_p\|) \quad \forall i = 1 \dots C$$
- 4) Compute the error term: $ERR = \|W_{i^*} - X_p\|$.
- 5) Grow the network: if $ERR > GT$ and i^* is a boundary class-unit, create new units from i^* such that its neighbourhood is completely filled. Then go to step 7.
- 6) Update network: if $ERR < GT$ update the existing prototypes using the same formula as in the standard algorithm (step 4 from Box 1).
- 7) Update the learning rate $\eta(t+1)$ and the neighbourhoods $N_i(t+1)$, $i = 1 \dots C$.
- 8) Repeat steps 2-7 until all the patterns have been presented to the network and the specified number of epochs have elapsed.

3. GROWING FUZZY SELF ORGANIZING ALGORITHM WITH AFFINITY CONTROL STRATEGY

The GSOM algorithm and the FSOM technique solve each of the above difficulties separately. However, the two algorithms can not be used simultaneously because the application of the FSOM technique needs global information from the whole set of prototypes and patterns, while the GSOM

algorithm may create additional class-units, hence new prototypes, each time a different pattern is presented to the network.

To overcome this difficulty a new self organizing model is proposed, that combines the FSOM technique with the GSOM algorithm and a growing control strategy based on affinities between patterns and prototypes. The control strategy was designed to take into account the final result of the classification process, namely the TLPs associated to consumers from distribution networks. Basically, any load profiling procedure should produce a set of TLPs that must obey two rules: (i) each TLP should be distinctly different from the others, and (ii) each TLP should represent a relatively homogenous group of consumers. When applied with SOMs, the first rule means that the prototypes of the class-units must be sufficiently different one another. The second rule means that the original metered load profiles used to generate the prototype must be highly similar among them.

The application of these rules generate opposite results. Thus, few class-units will generate largely different TLPs (first rule will be obeyed), but the metered load profiles associated to each TLP would not be similar (second rule will be violated). On the contrary, too many class-units will generate TLPs obeying the second rule (very similar metered load profiles), but violating the first rule (some TLPs are very similar, corresponding actually to the same consumer category). The most adequate number of class-units will result as a compromise between the two rules.

The above two rules can be quantified using affinities between TLPs (prototypes in the SOM network) and metered load profiles (patterns in the SOM network). Two types of affinities can be considered here: (i) affinities between prototypes/TLPs, $A_{i,j}^{T-T}$ and (ii) affinities between a prototype/TLP and the patterns / metered load profiles associated to it, $A_{i,p}^{T-M}$:

$$\begin{aligned} A_{i,j}^{T-T} &= \sqrt{\|W_i - W_j\|} = \sqrt{\sum_{k=1}^{NK} \alpha_k \cdot (w_{i,k} - w_{j,k})^2}; \\ A_{i,p}^{T-M} &= \sqrt{\|X_p - W_i\|} = \sqrt{\sum_{k=1}^{NK} \alpha_k \cdot (x_{p,k} - w_{i,k})^2}, \end{aligned} \quad (1)$$

where $i, j = 1 \dots NC$, $p = 1 \dots NP$, and $Class(p) \equiv i$. As stated above, it is desirable that affinities $A_{i,j}^{T-T}$ should have greater values, while affinities $A_{i,p}^{T-M}$ should have smaller values. At a global level, equations (1) can be used to define affinity degrees, namely: (i) global affinity degree between prototypes / TLPs – Deg^{T-T} and (ii) global affinity degree between a prototype / TLP and patterns / metered load profiles – Deg^{T-M}

$$\begin{aligned}
Deg^{T-T} &= \frac{2}{NC \cdot (NC - 1)} \cdot \sum_{i=1}^{NC-1} \sum_{j=1}^{NC} A_{i,j}^{T-T}; \\
Deg^{T-M} &= \frac{1}{NC} \cdot \sum_{i=1}^{NC} \sum_{p=1}^{NP} \left(A_{i,p}^{T-M} \left/ \sum_{p=1}^{NP} \delta_{i,p} \right. \right).
\end{aligned} \tag{2}$$

Equations (1) and (2) uses the following notations: NK -number of components in a pattern or prototype; NC -number of class-units; NP -number of patterns; $w_{i,k}$ – the k -th component of the prototype associated to class-unit i ; $x_{p,k}$ – the value of the k -th component of pattern p ; α_k – weighting coefficient for the k -th component of a prototype / pattern; $\delta_{i,p}$ – membership degree of pattern p to class i ($\delta_{i,p}=1$ if pattern p is associated to class i , and $\delta_{i,p}=0$ otherwise).

The weighting coefficients α_k in eq. (1) are used to assess the weight of load values at different moments in the day. Hence, coefficients α_k are computed using values K_P and K_V for the peak and valley load hours respectively ($K_P, K_V > 1$), based on weighting windows of lengths dt_P and dt_V around peak (T_P) and valley (T_V) hours. More details can be found in [4].

As to the growing mechanism used by the GSOM algorithm from Box 2, one can observe that this model is best suited for mapping problems, for which the growing process adds new units in the neighbourhood of the winning class-unit. However, for classification problems new units must be added one at a time with no need to consider any topological information. Hence, the GSOM algorithm from Box 2 will be applied using a simple growing strategy: simply add a new class-unit, and associate the new pattern to its prototype. Based on the above mechanisms, a new growing fuzzy self organizing algorithm with affinity control strategy (GFSOM-ACS) is proposed, as described in Box 3.

In this algorithm, the control function $Ctrl$ is computed as a weighted sum of the two global affinity degrees from eq. (2). The weighting coefficient β from Box 3 has values between 0 and 1. In our tests β was chosen to be 0.5, i.e. equal weights for affinity degrees Deg^{T-T} and Deg^{T-M} . During the evolution phase (while the SOM network is growing), Deg^{T-T} has increasing values, while Deg^{T-M} has decreasing values. Since $Deg^{T-T} \gg Deg^{T-M}$, the control function will tend to increase and the growing process must continue, adding a new class-unit, and incrementing the iteration count (first case for step 11, in Box. 3). At the end of the growing process, Deg^{T-T} begins to decrease and Deg^{T-M} begins to increase. The control function $Ctrl$ begins to decrease and the growing process must cease (second case for step 11, in

Box 3). The solution for the load profiling problem should be considered the one described by the SOM structure in the previous iteration, before the performances begun to depreciate (step 12, in Box 3).

4. IMPLEMENTATION OF THE GFSOM-ACS ALGORITHM

The algorithm described in Box 3 was tested using a load data collection consisting of 96 load profiles metered on the low voltage side of substation transformers supplying energy to consumers with a known structure [3]. Before processing, load profiles were transformed from kW to p.u. using as reference the average value of the daily load. This database was used in a previous study

Box 3

The GFSOM-ACS algorithm

- 1) Initialize the network structure: number of class-units NC , iteration count t and control function $Ctrl^*$.
- 2) Excepting the first iteration, save the solution from the previous iteration: $NC^* = NC - 1$, and $W^* = W(t-1)$, and $Ctrl^* = Ctrl$.
- 3) Initialize network parameters: prototypes $W_i(t)$ ($i=1 \dots C$), maximum number of epochs T_{max} , FSOM model parameters (m and m_0).
- 4) Present all patterns X_p ($p=1 \dots NP$) to the network and compute learning rates η_{ip} .
- 5) Update network prototypes $W_i(t)$ using :

$$W_i(t+1) = W_i(t) + \eta_{i,p} \cdot \Delta W_{i,p}$$
- 6) Repeat steps (4) – (5) until the maximum number of epochs is reached, then proceed to step 7.
- 7) Compute affinities between prototypes, $A_{i,j}^{T-T}$ from eq. (1) – affinities to be maximized.
- 8) Compute affinities between prototypes and their associated patterns, $A_{i,p}^{T-M}$ from eq. (1) – affinities to be minimized.
- 9) Compute global affinity degrees Deg^{T-T} and Deg^{T-M} using eq. (2).
- 10) Compute the control function:

$$Ctrl = \beta \cdot Deg^{T-T} - (1-\beta) \cdot Deg^{T-M}.$$
- 11) Iteration control:
 - If $Ctrl > Ctrl^*$, increment number of prototypes / classes ($NC = NC + 1$) and iteration count ($t = t+1$) and return to step 2.
 - If $Ctrl < Ctrl^*$, stop iterations and proceed to step 12.
- 12) The load profiling solution is described by the structure of the Kohonen network in the previous iteration: NC^* and W^* .

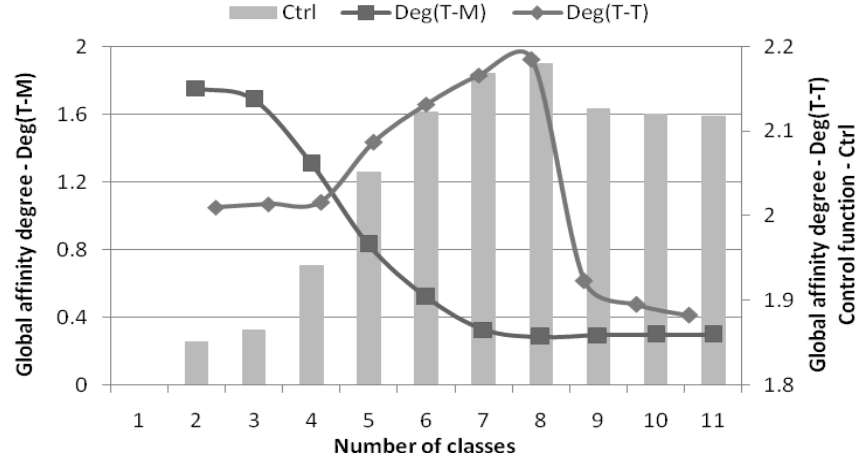


Fig. 1 – Global affinity degrees and the control function during the growing phase.

conducted to generate TLPs using the FSOM technique, with weighting windows around peak and valley load hours [3]. The optimal values for parameters used by the FSOM model were determined in this study. The same values, i.e. ($m = 2.00$, $m_0 = 100$, $dt_P = 0$, $dt_V = 1$, $K_P = 5$ and $K_V = 5$), were used in the present analysis.

The GFSOM-ACS algorithm was applied starting from 2 class-units in the SOM network. The growing process continued until the 9-th class-unit was added to the SOM network. The evolution of the global affinity degrees between prototypes / TLPs – Deg^{T-T} – and between prototypes and patterns / metered load profiles – Deg^{T-M} – are shown in Fig. 1, which contains data produced partially during the run of GFSOM-ACS algorithm (from 2 to 9 classes), and partially by an off-line simulation of the FSOM algorithm using 10 and 11 classes. As one can see from Fig. 1, the best value of the control function was obtained for 8 classes (see the bar diagram in Fig. 1). At the same time, the global affinity degree between prototypes Deg^{T-T} reach a maximum value, and the global affinity degree between prototypes and patterns Deg^{T-M} reach a minimum value in the same position, namely 8 classes.

Comparing this result with previous classification results from paper [3], one can see that the GFSOM-ACS algorithm produces 8 classes, while the basic FSOM algorithm produces 9 classes. In fact, 7 of these classes are the same and have the same TLPs in both cases. The difference consists in the fact that, while the FSOM algorithm produces 2 TLPs for two groups of relatively similar residential consumers, the new algorithm combines both groups in a unique TLP (TLP “Residential 2” from Fig. 2). In fact, the affinity between the two distinct TLPs

generated by the FSOM algorithm in [3] equals 0.42, a small value compared to the values of the Deg^{T-T} parameter (between 1.9 and 2.2) in GFSOM-ACS algorithm. The 8 TLPs generated by the application of the GFSOM-ACS algorithm to the load profiles data base are presented in Fig. 2.

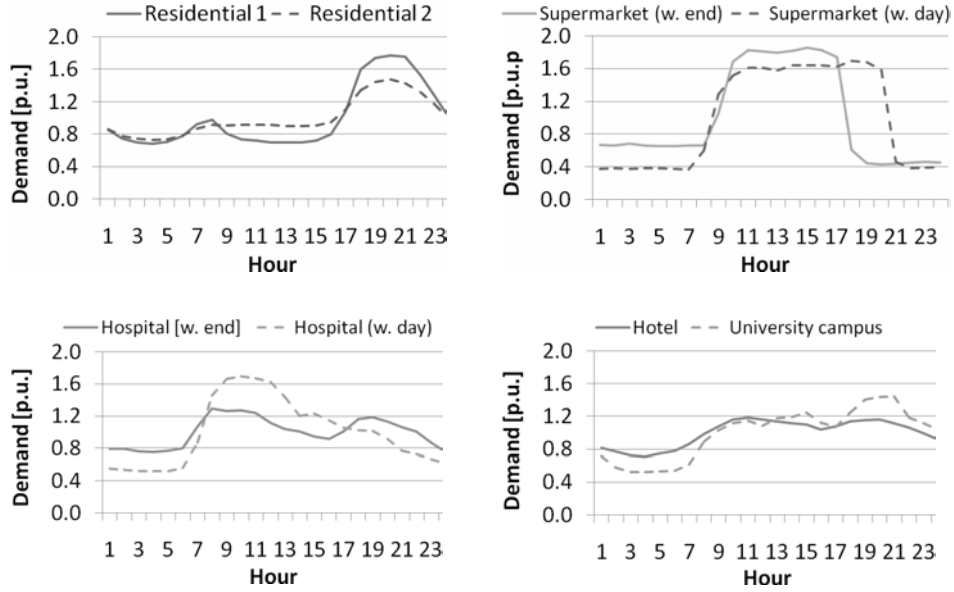


Fig. 2 – TLPs generated by the GFSOM-ACS algorithm.

5. CONCLUSIONS

This paper describes an approach to the consumer category classification according to the shape of their load profiles and the extraction of the TLPs for each category. The classification approach is based on a modified growing fuzzy-SOM algorithm with affinity control strategy.

The algorithm was tested using a learning data set consisting of 96 load profiles metered on the low voltage side of substation transformers supplying energy to consumers with a known structure. The results show a good behaviour of the proposed algorithm in terms of robustness and stability with respect to the structure of the data base used in the learning stage.

Received on 17 February, 2008

REFERENCES

1. R. Amarasari, D. Alahakoon, K. Smith, *Applications of Growing Self Organizing Map on High Dimensional Data*, In Proc. of the International Information Technology Conference (IITC), pp. 169–174, Colombo, Sri Lanka, 2004.
2. A. Baralady, P. Blonda, *A Survey of Fuzzy Clustering Algorithms for Pattern Recognition*, IEEE Trans. On Systems, Mna and Cybernetics – Part B, **6**, pp. 778–785 (1999),.
3. M. Gavrilas, C. V. Sfintes, M. N. Filimon, *Identifying typical load profiles using neural-fuzzy models*, In Proc. of 2001 IEEE/PES Transm. and Distrib. Conf. and Exposition, pp. 421–426, Atlanta, 2001.
4. M. Gavrilas, O. Ivanov, C. V. Sfintes, *Enhanced particle swarm optimization method for power loss reduction in distribution systems*, In Proc. of 19th International Conference on Electricity Distribution CIRED, Paper No 0088, CD-ROM, Viena, 2007.
5. H. Lee Willis, *Spatial electric load forecasting*, M. Dekker Inc., New York, 1996.
6. C. G. Looney, *Pattern recognition using neural networks – Theory and algorithms for engineers and scientists*, Oxford University Press, New York, 1997.
7. K. Mizutani, S. Miyamoto, *Kernel Based Fuzzy Competitive Learning Clustering*, In Proc. of the 14th IEEE Internl. Conf. On Fuzzy Systems, Reno-USA, 2005, pp. 636–639.