# Prediction of chaotic time-series using dynamic cell structures and local linear models

Lucius Chudý, Igor Farkaš *
Institute of Measurement Science
Slovak Academy of Sciences
Dúbravská cesta 9, 842 19 Bratislava, Slovakia
e-mail: {chudy,farkas}@neuro.savba.sk

**Abstract**

We present a time-series predicton method based on the combination of an unsupervised growing neural network – Dynamic Cell Structures (DCS) and local linear models (LLMs). DCS is used for representation of the attractor of the underlying dynamical system in the form of directed graph and thus provides the proper quantization of the state space data. Whereas such a model provides a highly accurate prediction of "simple" data (e.g. Mackey-Glass chaotic data), for data which exhibits so called dificult mappings and/or training set is not representative enough (e.g. laser data) it suffers from rather poor generalization. We analyse possible reasons for this behavior and propose a regularization technique which is optimized on cross-validation set to improve generalization. We also investigate unit insertion strategies intended to improve the accuracy of LLMs. Finally, we compare our results with related approaches.

## 1 Introduction

Prediction of (chaotic) time-series belongs to difficult information tasks, but enjoys the considerable attention, probably due to its high practical importance. Throughout the last decades, there have been developed two classes of prediction models, based on either conventional methods (linear predictions, polynomial approximations etc.), or more powerful neural networks [1], [2]. Global models attempt to yield a compact representation of an underlying dynamical system and require relatively few tunable parameters. On the other hand, local models lack the property of compactness and are not so economical, but have the ability to adhere to the local shape of an arbitrary surface, which is difficult especially in cases when the dynamical system characteristics vary considerably throughout the state space. Therefore, in a number of cases local models outperform global ones [3].

Local models are typically based on the concept of nearest neighbors. Building local mappings in all points of reconstructed state space is a time- and memory-consuming process, which led to a natural idea to quantize the state space and build local mappings only in positions given by obtained prototype vectors (Voronoi centers). There have been various local methods proposed, ranging from linear to high-order nonlinear models. As Farmer and Sidorowich [4] have already shown, local linear models (LLMs), despite their simplicity, provide an effective and accurate approximation, while high-order polynomials in higher dimensions are not significantly better than the first-order ones.

For quantization of the state space, in most of the neural networks papers, the self-organizing map (SOM) [5] has been used [6], [7], [8]. Apart from quantization, the SOM attempts to represent the topology of the approximated data manifold, so it can serve as a visualization tool as well. On the other hand, predefined map structure is a disturbing factor in quantization if the map dimension does not coincide with the intrinsic dimension of data. Higher approximation flexibility is thus offered by "structure-free" self-organizing network models, either with predefined number of units (prototypes) such as Lloyd and MacQueen's K-means clustering procedure [9], [10], the neural-gas network [11] or, time-varying, incremental neural network structure [12], [13], [14], [15].

In this paper, we use the dynamic cell structures (DCS) [15] for incremental quantization of the state space which is combined with LLMs. This type of unsupervised neural network grows during learning (quantization), until the predefined stopping criterion is met.

*This work was supported by Slovak Grant Agency for Science, grants No. 2/5088/98 and No. 98/5305/468.

However, such a type of prediction approach suffers from some inherent drawbacks. First, the quantization process may result in clusters containing patterns with different and conflicting "tangents", which implies a poor LLM. Second, a difficulty may arise if small clusters emerge. Third, associated with state space representation, so called difficult mappings may occur in the traning set (close input patterns map to distant outputs). Fourth, the problem is if the training set is not representative.

We investigate two approaches which aim at minimizing some of these difficulties. The first one deals with possible strategies of exploiting the prediction error during quantization for improvement of the consistency of LLMs. The second one regularizes the LLMs using the cross-validation set in order to improve generalization.

# 2 Dynamic cell structures

DCS [15] belong to the category of topology-representing networks, which were previously introduced by Martinetz [16]. DCS network can be viewed as an undirected graph $\mathcal{G} = (W, C)$, where $W = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n\}$ is the set of $n$ nodes (units) given by their coordinates, $\mathbf{w}_i \in \mathcal{R}^D$, with $D$ being the dimension of input space, and $C[n \times n]$ is the (symmetric) connection matrix defining edges in $\mathcal{G}$. $C_{ij} = C_{ji} \geq 0$ defines the strength of the connection between units $i$ and $j$. Shortly, DCS can be viewed as a topology-preserving vector quantizer.

Learning strategy for DCS consists of sequential insertion of units into $\mathcal{G}$, one after each sweep, until the stopping criterion is met (e.g. number of units, quantization error). Sweep is a number ($\lambda$) of iterations, each of which comprises a set of network adaptation steps. In each iteration, first, an input is chosen from the training set, $\mathbf{x} \in \mathcal{X} \subset \mathcal{R}^D$. Second, the weights of the Euclidean winner and its neighbors are updated using Kohonen-like rule (with constant learning rates $\epsilon_B, \epsilon_N$, respectively). Third, all connections are modified: the one between two winners is set to 1, connections falling below the threshold $\theta$ are set to zero and all remaining ones are multiplied by $0 < \alpha < 1$. Recommended relation in off-line tasks is $\alpha = \sqrt[|\mathcal{X}|]{\theta}$, which presumes setting $\lambda = |\mathcal{X}|$, where $|\mathcal{X}|$ denotes the size of the training set.

Finally, after the sweep, a new unit is inserted into DCS according to units' resources which quantify local approximation errors. It is placed (1) either to match the input pattern, which caused the largest error, or (2) in between the units with the largest and its neighbor with the second largest resource value. In the latter case, resources are proportionally redistributed among the participating three units. After unit insertion, new connections are set to 1, and resources are decremented (multiplied by $0 \leq \beta < 1$) in order to prevent their overflow. In off-line tasks, it is reasonable to nullify resources after each sweep.

As DCS are commonly intended for vector quantization, resources are used to quantify the quantization error (QE). In our task, however, the proper choice is to update resources based on local prediction error (PE), to achieve a better prediction model.

In addition, in both cases one can use either accumulated or average error (per input pattern). Note that using PE based insertion supposses an on-line approach, i.e. building LLMs during the quantization process after each sweep. As indicated in our experiments, this approach may improve the accuracy of LLMs due to "less conflicting" patterns within clusters.

# 3 Prediction method

## 3.1 State space quantization

In time-series prediction, the time-series is first embedded into the state space using delay coordinates

$$\mathbf{x}(t) = [x(t), x(t-\tau), \ ..., x(t-(d_E-1)\tau]^T \ , \tag{1}$$

where $x(t)$ is the value of the time-series at time $t$, $\tau$ is a suitable time delay, and $d_E$ is the embedding dimension. Vector $\mathbf{x}(t)$ is used to predict the value $x(t+\Delta t)$. Input-output vectors $\mathbf{z}$ (used for training) are formed by concatenation as

$$\mathbf{z}(t) = [\mathbf{x}^T(t), x(t+\Delta t)]^T \ .$$

Every unit $i \in \mathcal{G}$ has a vector $[\mathbf{w}_i^T(t), w_i(t+\Delta t)]^T \in \mathcal{R}^D$ associated with it, where its input part $\mathbf{w}_i(t) = [w_{i1}(t), ..., w_{id_E}(t)]^T \in R^{d_E}$. State space quantization can in principle be performed in two ways: (1) in input space (based on $\mathbf{x}$'s), or (2) in input-output space (based on $\mathbf{z}$'s). In both cases, however, the update of $\mathbf{w}_i(t)$ is performed in $R^D$, i.e. the input-output space. [1]

[1] The first case corresponds in fact to the weight adaptation strategy of constrained topological mapping [17].

The quantization process with DCS starts with two units (placed to be identical with first two inputs), and input-output vectors $\mathbf{z}$(t) are presented randomly with respect to time $t$. Unit insertion process is governed either by QE or PE. As a stopping criterion, we used the desired number of units.

### 3.2 Local linear models

Process of building LLMs consists of repetition of two main steps: winner search and parameter adaptation. Winner $B$ among units is searched comparing only the input parts both of input-output and weight vectors, i.e., $B = arg\min_i \|\mathbf{x}(t) - \mathbf{w}_i(t)\|$. Local prediction models for the winner and its neighbors yield the coefficients $\{\mathbf{a}_i \in \mathcal{R}^{d_E}\}, \forall i \in \mathcal{N}_B$, so that prediction in the neighborhood of winner $i$ is computed as

$$\hat{x}(t + \Delta t) = \mathbf{a}_i^T(\mathbf{x}(t) - \mathbf{w}_i(t)) + w_i(t + \Delta t) \ .$$

Prediction coefficients $\mathbf{a}_i$ are computed using standard singular value decomposition (SVD) procedure.

### 3.3 Regularization of LLMs

Using SVD for solving linear set od equations can be an advantage for regularization of the solution, especially in the case of difficult mappings and/or not representative learning set. Namely, it might be useful to relax the rigidity of the prediction parameters estimate obtained from the learning set and allow less precise estimate by means of regularization of the pseudoinverse matrix. If the matrix $\mathbf{X}$ is decomposed as $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, then elements of its *effective pseudoinverse* $\mathbf{X}^+$ are computed as

$$x_{ij}^+ = \sum_{k=1}^{m} \frac{v_{ik} u_{jk}}{\sigma_k} \ , \qquad i = 1, ..., d_E \quad j = 1, ..., n_i \ , \tag{2}$$

where $n_i$ is the size of $i$-th Voronoi cluster. Usually, only elements corresponding to relatively very small singular values $\sigma_k$ are omitted, in order to eliminate noise and numerical errors in computation. Our motivation is to improve generalization, so we allow more elements to be ommitted in (2). However, optimal $m$ is difficult to estimate a priori. We determine this number locally, i.e. separately for every cluster by means of minimizing the PE on the cross-validation set (previously extracted from $\mathcal{X}$). It is accomplished by trial and error, i.e. simply by examining local solutions with various $m$ and choosing the one which yields the smallest PE.

## 4 Experiments

First, for the purpose of comparison with other methods, we tested our algorithm on a benchmark time-series, generated by the Mackey-Glass system [18], which is described by the differential equation $x'(t) = -0.1x(t) + 0.2x(t-\delta)/(1 + x(t-\delta)^{10})$, where $x(t)$ is the value of the time-series at time $t$. We used data available from CMU archive [19], in which case $\delta = 17$. For creating the embedded data vectors for training and testing, standard values $\tau = 6, d_E = 4$ were used (in Eq. 1).

The second test set was laser data in chaotic regime downloaded from [20]. In this case, we chose $\tau = 1, d_E = 15$. Prediction accuracy was evaluated by normalized root mean squared error (NRMSE).

### 4.1 Results

**Mackey-Glass data.** First, the performance of the proposed method was tested on various sizes of the DCS graph, using the final number of units $n_f$ as a stopping criterion. After quantization of the state space, LLMs were built. Results are summarized in Table 1. Parameters of the DCS were set by hand as follows: $\alpha = 0.999, \lambda = 200, \beta = 0, \theta = 0.001, \epsilon_B = 0.2, \epsilon_N = 0.01$, $|\mathcal{X}| = 3000$, number of iterations for building LLMs: one pass over $\mathcal{X}$. Inputs were taken in natural order, consecutively with respect to time $t$. The DCS was trained to predict $x(t + 6)$. In order to predict $x(t + 84)$, the predicted values $\hat{x}(t + 6)$ were fed back iteratively 14 times. Also, in order to see the growth of prediction error in autonomous regime, we included the *evol* prediction mode, i.e. the network was supplied with the first input vector of the test set, and the rest of it was estimated during the next (95) iterative predictions. As for insertion strategies, both QE and PE worked equally well, so we present only QE based one, which is faster.

For training, also smaller sets were tried in order to obtain results for comparison with related approaches. Specifically, several runs were performed, each using a different 500- or 1000-fraction of $\mathcal{X}$.

| Number | $x(t+6)$ | | | $x(t+84)$ | | | evol | | |
|---|---|---|---|---|---|---|---|---|---|
| of units | 500 | 1000 | 3000 | 500 | 1000 | 3000 | 500 | 1000 | 3000 |
| 100 | 0.015 | 0.013 | 0.0135 | 0.048 | 0.062 | 0.064 | 0.20 | 0.52 | 0.420 |
| 200 | 0.010 | 0.009 | 0.0055 | 0.042 | 0.048 | 0.030 | 0.15 | 0.29 | 0.085 |
| 400 | - | - | 0.0033 | - | - | 0.018 | - | - | 0.082 |

Table 1: NRMSE of prediction for 3 different sizes of DCS and 3 sizes of the Mackey-Glass training set. SVD was used for building LLMs.

**Laser data.** Quantization process based on QE may in some cases lead to improper partitioning of the state space with regard to building LLMs. This does not appear to be a problem in case of simple Mackey-Glass signal, but seems to matter in the case of more difficult laser data. We investigated the strategy of unit insertion based on PE. Specifically, a new unit is inserted in the place of a pattern which has caused the largest PE within the cluster with largest PE. To make comparison, we tried also QE based insertion strategy. Both were combined with two basic unit positions updates – finding a winner either in input or input-output space. In Fig. 1 are shown 4 averaged (over 20 random runs) developments of NRMSE on $\mathcal{X}$ (3000 samples) for above 4 cases. In all cases, DCS developed on QE basis, until 100 neurons were inserted (in order to avoid building LLMs for initially very large clusters). This configuration served as a starting configuration for subsequent comparison. As can be seen, insertion strategies based on PE speed up learning and this process fluctuates less in comparison to QE based strategies.

*Figure 1 to be inserted here.*

Figure 1: Averaged NRMSE obtained on training sets of laser data. IN (OUT) stands for quantization performed in input (input-output) space, P (Q) stands for PE (QE) based unit insertion. It is evident that PE based insertion converges faster and is more stable.

*Regularization.* Although investigated PE insertion strategies lead to the decrease of NRMSE on $\mathcal{X}$ (see Fig. 1), they can have minor effect on PE on the test set $\mathcal{Y}$ (poor generalization, especially for collapses [21]), mainly in the case of unrepresentative $\mathcal{X}$. This seems to be the case also for laser data when even for the best PE insertion strategy the NRMSE on $\mathcal{Y}$ considerably fluctuates and does not decrease during training. Proposed regularization of every LLM minimizes the PE on cross-validation set and indirectly also decreases PE on $\mathcal{Y}$. The final solution was chosen to minimize the weighted sum of PE on cross-validation (by a factor of 0.8) and training sets (0.2) during the growth of DCS. In Table 2 we present means and standard deviations of NRMSE evaluated on $\mathcal{Y}$ during the growth of DCS from 100 to 150 neurons for an unregularized LLMs, as well as two regularizations (corresponding to sizes of cross-validation sets equal to 1/5 and 1/10 fractions of $\mathcal{X}$). It can be seen that whereas unregularized solution leads to a fluctuating poor result, regularizations yield much better generalization.

| NRMSE | Reg-0 | Reg-5 | Reg-10 |
|---|---|---|---|
| Mean | 0.485 | 0.150 | 0.140 |
| StDev | 0.577 | 0.047 | 0.027 |

Table 2: Means and standard deviations of NRMSE on laser data test set for unregularized LLMs and two regularized solutions. Averaged over 20 random runs.

## 4.2 Comparison to other methods

Using Mackey-Glass time-series as a benchmark data set, one typically attempts to predict $x(t+84)$ or $x(t+85)$. In the former case, iterative prediction based on $\hat{x}(t+6)$ is usually applied.

To make comparisons compatible, we focus only on neural networks trained on Mackey-Glass time-series with $\delta = 17$. Among the approaches based on self-organized quantization of the state space and local linear modelling, we include SOM+SVD [8] and NGN+ALLM [2] [11] methods.

[2] ALLM stands for adaptive LLMs, which is an incremental adaptation of parameters derived from least-squares error minimization.

As a best result, the NRMSE for SOM+SVD was achieved in configuration 1225 units/3000 inputs, namely {0.0048,0.022} for predicting $\{x(t+6), x(t+84)\}$. For smaller maps, errors for 100/3000 were {0.013,0.06} and for 400/3000 {0.01,0.14}. For smaller training sets, 1225/500 resulted in {0.033,0.12} and 1225/1000 in {0.009,0.035}. If we consider our results in configuration 400/3000, given in Table 1, we can see that DCS+SVD achieves roughly the same NRMSE with 3 times fewer units than SOM. This must be due to higher flexibility of DCS in approximating (partitioning) nonlinear data manifold.

Neural gas network (NGN), alleviates the the above problem, because it is not restricted to any a priori topology either. Reported results for prediction of $x(t+90)$ are approximately as follows: [3] NRMSE in configurations $n_f/*$, where $*$ stands for {500,1000,5000} inputs, resulted in {0.266,0.224,0.211}, for $200/*$ in {0.251,0.188,0.158} and for $400/*$ in {0.298,0.193,0.133}. Since we assume that both DCS and NGN are comparable in their quantization performance, it can be observed that ALLM is less accurate in prediction compared to SVD.

Other neural methods, which have been used for predicting Mackey-Glass time series are back-propagation neural network (BPNN) [1], and RAN [22] based on radial basis functions. With BPNN having 20 hidden units in a two-layer architecture and training set size 500, the authors reported NRMSE of {0.02,0.06}. RAN trained on 3000 inputs produced the error 0.054, but for direct prediction of $x(t+85)$.

Regarding laser data, it is difficult to compare our results with known results from the literature, since they involve sophisticated preprocessing of primary data and/or high complexity of computation. Sauer's approach [23] involves spline interpolation, low-pass filtering of data, PCA projection, weighted regression and a huge number of LLMs even for various time prediction horizons. Wan's method [24] is based on a quite sophisticated and complex FIR feedforward network model with 25-dimensional input representation and complicated architecture comprising 3 successive layers. The method used in [25] involves weighted PCA, expert combination of linear models and multi-layer perceptrons and adjustable prediction horizon. The main goal of our approach was to investigate the prediction capabilities of a rather simple model requiring no preprocessing of data. Despite that our results do not compare favourably with the above mentioned approaches, proposed modifications of the basic DCS-LLM method appear to be successful in improving a very poor prediction on laser data to the acceptable accuracy.

# 5 Conclusion

Our results strongly indicate that further improvement of the prediction accuracy of the proposed DSC-LLM model needs a proper preprocessing of primary data either by means of filtering, interpolation and/or projection of data. These seem to be the crucial factors which influence prediction generalization. A possible improvement of the DCS-LLM model itself may result from incorporation of the distance to the subspace of particular LLM for finding the "nearest" unit. Such a combination of Euclidian distance and LLM subspace distance might lead to better partitioning of data and thus to more consistent LLMs.

# References


[1] A. Lapedes and R. Farber. Nonlinear signal processing using neural networks: Prediction and system modelling. Technical Report LA-UP-87-2662, Los Alamos National Laboratory, 1987.

[2] S. Haykin. Neural networks expand SP's horizons. *IEEE Signal Processing Magazine*, pages 24–49, March 1996.

[3] A. C. Singer, G. W. Wornell, and A. V. Oppenheim. Codebook prediction: A nonlinear signal modeling paradigm. In *Proc. of ICASSP'92*, volume V, pages 325–328, 1992.

[4] J. D. Farmer and J. J. Sidorowich. Predicting chaotic time series. *Physics Review Letters*, 59(8):845–848, 1987.

[5] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, 1997. second edition.

[6] J. Walter, H. Ritter, and K. J. Schulten. Non-linear prediction with self-organizing maps. In *Proc. of the IJCNN-90*, volume 2, pages 589–594, 1990.

[7] J. C. Principe and L. Wang. Non-linear time series modelling with self-organizing feature maps. In *Proc. of the 1995 Workshop: Neural Networks for Signal Processing V.*, pages 11–20, 1995.

[8] J. Vesanto. Using the SOM and local models in time-series prediction. In *Proc. of the WSOM'97: Workshop on Self-Organizing Maps*, pages 209–214. Espoo, Finland, 1997.


[3] The values of NRMSE were estimated from the line graph, given in their paper, displaying log(NMSE).

[9] S. P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Berkeley Symposium on Mathematics, Statistics, and Probability*, volume I, pages 281–297. Berkeley: University of California Press, 1967.

[11] T. Martinetz, S. G. Berkovich, and K. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.

[12] B. Fritzke. Growing cell structures - a self-organizing network for unsupervised and supervised learnig. *Neural Networks*, 7(9):1441–1460, 1994.

[13] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, 1995.

[14] J. Blackmore and R. Miikkulainen. Incremental grid growing: encoding high-dimensional structure into a two-dimensional feature map. In *Proc. of ICNN'93, San Francisco, CA*, pages I–450, 1993.

[15] J. Bruske and G. Sommer. Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, 7:845–865, 1995.

[16] T. Martinetz. Topology representing networks. *Neural Networks*, 7:505–522, 1994.

[17] V. Cherkassky and H. Lari-Najafi. Constrained topological mapping for nonparametric regression analysis. *Neural Networks*, 4(1):27–40, 1991.

[18] M. C. Mackey and L. Glass. Oscillations and chaos in physiological control systems. *Science*, 197:287–289, 1977.

[19] http://legend.gwydion.cs.cmu.edu/neural bench/benchmark/mackey-glass.html.

[20] http://www.cs.colorado.edu/ andreas/Time-Series/SantaFe.html.

[21] A. S. Weigend and N. A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume XV, Reading, MA, 1994. Santa Fe Institute, Addison-Wesley.

[22] C. J. Platt. A resource-allocating network for function interpolation. *Neural Computation*, 3(2):213–225, 1991.

[23] T. Sauer. Time series prediction by using delay coordinate embedding. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume XV, pages 175–193, Reading, MA, 1994. Santa Fe Institute, Addison-Wesley.

[24] E. Wan. Time series prediction by using a connectionist network with internal delay lines. In A. S. Weigend and N. A. Gershenfeld, editors, *Time Series Prediction: Forecasting the Future and Understanding the Past*, volume XV, pages 195–217, Reading, MA, 1994. Santa Fe Institute, Addison-Wesley.

[25] R. Bakker, J. C. Schouten, C. M. van den Bleek, and C. Lee Giles. Neural learning of chaotic dynamics: the error propagation algorithm. Technical Report UMD-CS-TR-3843, NEC Research Institute, 1997.