Neural Network Analysis of Russian Parliament Voting Patterns

Dusan Husek
Acad. of Sci. of the Czech Republic, Institute of Computer Science, the Czech Republic
Email: dusan@cs.cas.cz

Alexander A. Frolov
IHNA Russian Acad. of Sci., Russia
Email: aafrolov@mail.ru

Pavel Y. Polyakov
IONT Russian Acad. of Sci., Russia
Email: pavel.mipt@mail.ru

Hana Rezankova
University of Economics, Prague, the Czech Republic
Email: rezanka@vse.cz

ABSTRACT

The recurrent neural network capable to provide the Boolean factor analysis of the binary data sets of high dimension and complexity is applied to discovery of voting patterns in the Russian parliament. The new method of sequential factor extraction based on the Lyapunov function is discussed in deep. Efficiency of the new method is shown on simulated data and on real data from Russian parliament as well.

Key Words: Neural networks, associative memory, recurrent neural network, Boolean factor analysis, clustering, data mining.

1. Introduction

Theoretical analysis and computer simulations performed in [3] revealed Hopfield-like neural networks capability of performing the Boolean factor analysis of signals of high dimension and complexity. Factor analysis is a procedure which maps original signals into the space of factors. The principal component analysis (PCA) is a classical example of such a mapping in the linear case. Linear factor analysis implies that each original N-dimensional case can be presented as

$$X = \mathbf{F} \otimes S \oplus \varepsilon \qquad (1)$$

where $\mathbf{F}$ is a matrix $N \times L$ of factor loadings, S is a L-dimensional vector of factor scores and ε is an error. (Please mention that statisticians often use another notation equivalent to mentioned above $X^{\mathbf{T}} = S^{\mathbf{T}} \otimes \mathbf{F}^{\mathbf{T}} \oplus \varepsilon^{\mathbf{T}}$, this incomes from tabular format of source data – observation – where $X^{\mathbf{T}}$ is row in the table). Here each component of S gives contribution of a corresponding factor in the original signal. Columns of loading matrix $\mathbf{F}$ give vectors presenting corresponding factors in the signal space. In the following namely these vectors are termed factors. The mapping of the original feature space to the factor space means, that original signals are represented by vectors S of dimension, lower than the input signals X. Dimensionality of vectors S is much smaller than the dimensionality of signals X. Thereby the factor analysis provides high compression of original signals.

Boolean factor analysis implies that a complex vector signal has a form of the Boolean sum of weighted binary factors:

$$X = \bigcup_{\{i\}} \mathbf{F} \otimes S_{\{i\}}, \ |\{i\}| \le C . \qquad (2)$$

First this means that the original signals, factor scores and factor loadings are binary

and secondly that the mapping of the original signal to the factor space means discovery of factors that were mixed in the signal. The mean number of factors mixed in the signals we term signal mean complexity C_μ.

For the case of large dimensionality and complexity of signals it was a challenge [3] for us, to utilize the Hopfield-like neural network with parallel dynamics for the Boolean factor analysis. Binary patterns X of the signal space are treated as activities of N binary neurons (1 – active, 0 – nonactive) with gradually ranging synaptic connections between them. During the learning stage patterns $X^{(m)}$ are stored in the matrix of synaptic connections $\mathbf{J}'$ according to the Hebbian rule:

$$J'_{ij} = \sum_{m=1}^{M} (X_i^{(m)} - q^{(m)})(X_j^{(m)} - q^{(m)}), \quad i \neq j, \quad J'_{ii} = 0 \ , \tag{3}$$

where M is the number of patterns in the learning set and bias $q^{(m)} = \sum_{i=1}^{N} X_i^{(m)}/N$ is the total activity of the m-th pattern. This form of bias corresponds to the biologically plausible global inhibition being proportional to an overall neuronal activity.

Additionally to N principal neurons of the Hopfield network described above we introduced one special inhibitory neuron activated during the presentation of every pattern of the learning set and connected with all principal neurons by bidirectional connections. Patterns of the learning set are stored in the vector $\mathbf{J}''$ of the connections according to the Hebbian rule:

$$J''_i = \sum_{m=1}^{M} (X_i^{(m)} - q^{(m)}) = M(q_i - q), \tag{4}$$

where $q_i = \sum_{m=1}^{M} X_i^{(m)}/M$ is a mean activity of the i-th neuron in the learning set and q is a mean activity of all neurons in the learning set. It is also supposed that the excitability of the introduced inhibitory neuron decreases inversely proportional to the size of the learning set being $1/M$ after storing of all its patterns.

Due to the Hebbian learning rule (3), neurons which represent one factor and therefore tend to fire together, become more tightly connected than neurons belonging to different factors, constituting an attractor of network dynamics. This property of factors is a base of the proposed two-run procedure of factor search. Its initialization starts by presentation of a random initial pattern $X^{(in)}$ with $k^{(in)} = r^{(in)}N$ active neurons. The activity $k^{(in)}$ is supposed to be much smaller than the activity of all factors. After presentation of $X^{(in)}$ to neural network, its activity X evolves to an attractor. The evolution is determined by the parallel dynamics equation for discrete time. At each time step:

$$X_i(t+1) = \Theta(h_i(t) - T(t)), i = 1, \cdots, N, \quad X_i(0) = X_i^{(in)} \ , \tag{5}$$

where h_i are components of the vector of synaptic excitations

$$h_i(t) = \sum_{j=1}^{N} J'_{ij} X_j(t) - (1/M) J''_i \sum_{j=1}^{N} J''_j X_j(t), \tag{6}$$

Θ is a step function, and $T(t)$ is an activation threshold. The first term in (6) gives synaptic excitations provided by the principal neurons of the Hopfield network and the second one by the additional inhibitory neuron. The use of the inhibitory neuron is equivalent to the substraction of $(1/M)J''_i J''_j = M(q_i - q)(q_j - q)$ from the matrix J'_{ij}. Thus (6) can be rewritten as $h_i(t) = \sum_{j=1}^{N} J_{ij} X_j(t)$ where $\mathbf{J} = \mathbf{J}' - \mathbf{Mqq}^{\mathrm{T}}$, $\mathbf{q}$ is a vector with components $q_i - q$ and $\mathbf{q}^{\mathrm{T}}$ is a transposed $\mathbf{q}$. As shown in [3] the replacement of the common connection matrix $\mathbf{J}'$ by $\mathbf{J}$, first, completely suppressed two global attractors, which dominate in network dynamics for large signal complexity C, and second, made the size of attractor basins around factors to be independent of C.

At each time step of the recall process the threshold $T(t)$ was chosen in such a way that the level of the network activity was

kept constant and equal to $k^{(in)}$. Thus, on each time step $k^{(in)}$ "winners" (neurons with the greatest synaptic excitation) were chosen and only they were active on the next time step. To avoid uncertainty in the choice of winners, when several neurons had synaptic excitations at the given level of the activation threshold, a small random noise was added to the synaptic excitation of each individual neuron. The amplitude of the noise was put to be less than the smallest increment of the synaptic excitation given by formula (6). This ensured that neurons with the highest excitations were kept to be winners in spite of the random noise added to the neurons' synaptic excitations. The level of noise added to individual neurons excitation was fixed during the whole recall process to provide its convergence. As shown in [6], this choice of activation thresholds allows for stabilization of the network activity in point or a cyclic attractor of length two.

When the activity stabilizes at the initial level of activity $k^{(in)}$, $k^{(in)}+1$ neurons with maximal synaptic excitation are chosen for the next iteration step, and the network activity evolves to some attractor at the new level of activity $k^{(in)}+1$. Then the level of activity increases to $k^{(in)}+2$, and so on, until the number of the active neurons reaches the final level $r^{(f)}N$ with $r^{(f)} > p$. Here $p << 1$ (see [2]), is relative level of activity of the just revealing factor. Thus, one trial of the recall procedure contains $(r^{(f)} - r^{(in)})N$ external steps and several steps inside each external step to reach some attractor for a fixed level of activity.

At the end of each external step the relative Lyapunov function was calculated by formula

$$\Lambda = X^{\mathbf{T}}(t+1)\mathbf{J}X(t)/(rN), \qquad (7)$$

where $X^{\mathbf{T}}(t+1)$ and $\mathrm{X}(t)$ are two network states in the cyclic attractor (for a point attractor $X^{\mathbf{T}}(t+1) = X(t)$). The relative Lyapunov function is a mean synaptic excitation of neurons belonging to some attractor at the end of the external step with $k = rN$ neurons.

Attractors with the highest Lyapunov function would be obviously winners in most trials of the recall process. Thus, more and more trials are required to obtain a new attractor with a relatively small value of the Lyapunov function. To overcome this problem the dominant attractors should be deleted from the network memory. The deletion was performed according to the Hebbian unlearning rule by substraction $\Delta J_{ij}, j \neq i$ from synaptic connections J_{ij} where

$$\Delta J_{ij} = \frac{\eta}{2} J(X)[(X_i(t-1)-r)(X_j(t)-r)+ \\ +(X_j(t-1)-r)(X_i(t)-r), \qquad (8)$$

$J(X)$ is the average synaptic connection between active neurons of the attractor, $X(t-1)$ and $X(t)$ are patterns of network activity at last time steps of the iteration process, r is the level of activity, and η is an unlearning rate. Here we have to mention that for point attractors holds $X(t) = X(t-1)$, and for cyclic attractors $X(t-1)$ and $X(t)$ two states of attractor occur.

There are three important similarities between the described procedure of the Boolean factor analysis and the linear PCA. First, PCA is based on the same covariation matrix as the connection matrix in a Hopfield network. Second, factor search in PCA can be performed by the iteration procedure similar to that described by equations (5) and (6). The only difference is that the binarization of synaptic excitations using the step function must be replaced by their normalization: $X_i(t+1) = h_i(t)/|\mathbf{h}(t)|$. Then the iteration procedure starting from any random state converges to the eigenvector $\mathbf{f}_1$ of the covariation matrix with the largest eigenvalue Λ_1. Just this eigenvector is treated as the first factor in PCA. Third, to obtain the next factor, the first factor must be deleted from the covariation matrix by the substraction of $\Lambda_1 \mathbf{f}_1 \mathbf{f}_1^T$, and so on. The substraction is similar to Hebbian unlearning (8).

However, the Boolean factor analysis by

the Hopfield-like network has one principal difference from the linear PCA. Attractors of the iteration procedure in PCA are always factors while in Hopfield-like networks the iteration procedure can converge to factors (true attractors) and to spurious attractors which are far from all factors. Thus, two main questions arise from the point of view of the Boolean factor analysis by the Hopfield-like network. First, how often would network activity converge to one of the factors starting from a random state? Second, is it possible to distinguish true and spurious attractors when network activity converges to some stable state? Both these questions are answered in the next Section.

There are many examples of data in the sciences when the Boolean factor analysis is required [1]. In our previous papers [4, 5, 6] we used this neural network to analyze textual data. Here we apply our method to parliament voting data. The results are discussed in Section 3.

2. Artificial signals

To reveal peculiarities of true and spurious attractors we performed computer experiments with simulated data.

We generated factors in such a way that each one contained exactly $n = pN$ entries 1 and $(1-p)N$ entries zero. Thus, for each factor $\mathbf{f}^l \in B_n^N$, and analogously for each score $S \in B_C^L$ holds:

$$B_n^N = \left\{ X \mid X_i \in \{0,1\}, \sum_{i=1}^{N} X_i = n \right\}$$

Each pattern of the learning data set (observation) was generated to be a Boolean factor loadings superposition of exactly C factors. We supposed that factor loadings and factor scores are statistically independent (factors can overlap each other).

Then we analyzed Lyapunov function behavior during recall processes. As an example, Fig. 1 demonstrates changes of a relative Lyapunov function for $N = 3000$, $L = 5300$, $p = 0.02$ and $C = 10$. The recall process started at $r_{in} = 0.005$.

Trajectories of network dynamics form two separated groups. As shown in Fig. 2, the trajectories with higher values of the Lyapunov function are true and with lower ones are spurious. This Figure relates values of the Lyapunov function for patterns of network activity at points $r = p$ to maximal overlaps of these patterns with factors.

The overlap Ov between two patterns $X^{(1)}$ and $X^{(2)}$ with Np active neurons was calculated by formula

$$\mathrm{Ov}(X^{(1)}, X^{(2)}) = \frac{1}{Np(1-p)} \sum_{i=1}^{N} (X_i^{(1)} - p)(X_i^{(2)} - p)$$

According to this formula the overlap between two equal patterns is equal to 1 and the mean overlap between independent patterns is equal to 0. Patterns with a high Lyapunov function have high overlap with one of the factors, while the patterns with a low Lyapunov function are far from all the factors. It is shown that true and spurious trajectories are separated by the values of their Lyapunov functions. In Figs. 1 and 2 the values of the Lyapunov function are normalized by a mean value of this function over true attractors at the point $r = p$.

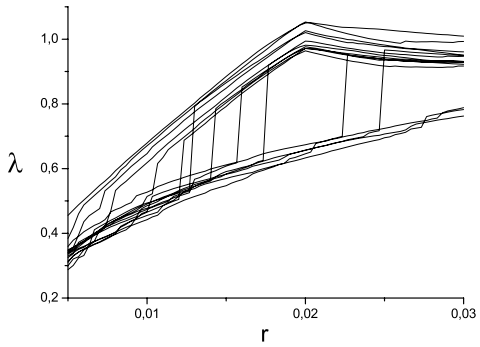

Fig. 1 *Relative Lyapunov function λ in dependence on the relative network activity r.*

The second characteristic feature of true trajectories is the existence of turning point at a point $r = p$ where the level of network activity coincides with that in factors (see Fig. 1). If $r < p$, then the increase of r results in almost linear increase of the relative Lyapunov function.

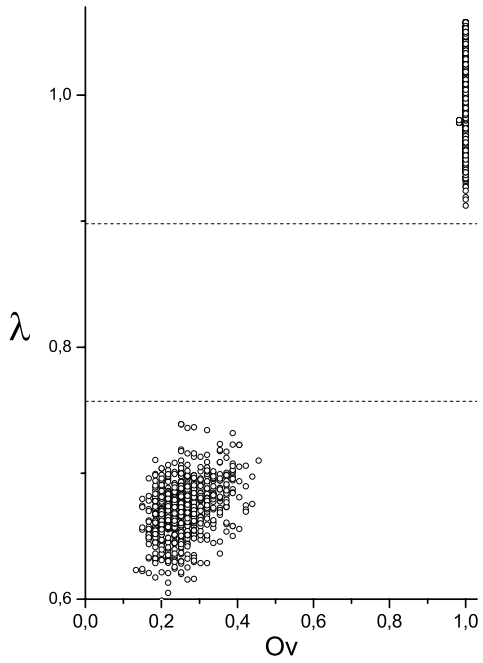

Fig. 2 *Values of normalized Lyapunov function in relation to overlaps with the closest factors.*

The increase of r occurs in this case due to the joining of neurons belonging to the factors that are strongly connected with other neurons of the factor. Then the joining of new neurons results in proportional increase of mean synaptic excitation to the active neurons of a factor that is just equal to their relative Lyapunov function. When $r > p$, the increase of r occurs due to joining of some random neurons that are connected with the factor by week connections. Thus, the increase of the relative Lyapunov function for a true trajectory sharply slows and it tends to the values of the Lyapunov function for spurious trajectories. The use of these two features of true trajectories provides reliable tool for recognition of factors. This phenomenon was clearly confirmed in our previous papers [4, 5, 6] where our method was used for textual data analysis. Here we apply neural network Boolean factor analysis technique to the analysis of parliament voting.

3. Analysis of parliament voting

For the following analysis we used as data source results of roll-call votes in the Russian parliament in 2004 [7]. Each vote is represented by a binary vector with component 1 if the correspondent deputy voted affirmatively and 0 negatively. The number of voting during the year was 3150. The number of deputies (consequently the dimensionality of the signal space and the network size) was 430 (20 deputies who voted less than 10 times were excluded from the analysis).

Fig. 3 shows the Lyapunov function trajectories starting from 1500 random initial states. One can distinguish only four trajectories as a result. Two of them have obvious turning points (kinks) and therefore were identified as two factors. The factor with the highest Lyapunov function consists of 50 deputies and completely coincides with the fraction of the Communist Party (CPRF).

Another factor consists of 36 deputies. All of them belong to the fraction of the Liberal-Democratic Party (LDPR) that has 37 chairs in the parliament in total. Thus one of the members of this fraction fell out of the corresponding factor. The pointed kinks at the corresponding trajectories give evidence that these fractions are the most disciplined and their members vote coherently.

Fig. 4 demonstrates trajectories after deleting of the two mentioned factors. 1500 runs of our algorithm starting from randomly selected initial network states resulted only into two trajectories in this case. On one of them we can see the turning point but it is not as strict as for CPRF and LDPR factors.

We hypothesized that the point where the second derivative of the Lyapunov function by k has minimum corresponds to the third factor. The factor consists of 37 deputies. All of them belong to the fraction "Motherland" (ML) which consists of totally 41 deputies. Thus 4 of its members fell out of the factor. The fuzziness of kink at the trajectory gives evidence that this fraction is not as homogeneous as the two first ones and actually the fraction split up in two fractions in 2005.

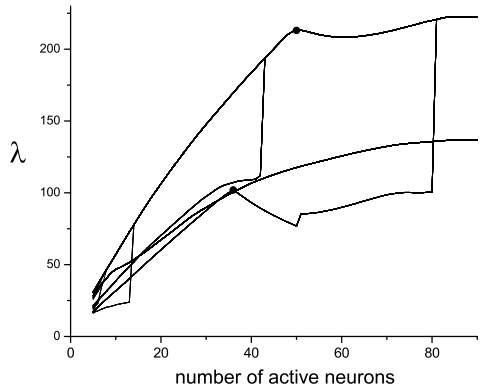

Fig. 3 *Relative Lyapunov function λ for parliament data in dependence on the number of active neurons. Thick points, on the trajectory, correspond to the first and the second factor.*

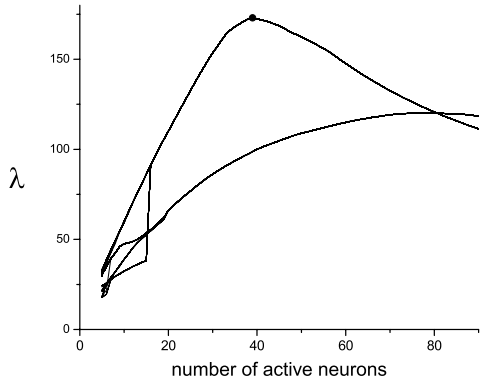

Fig. 4 *The same as in Fig. 3 after deleting two first factors. The thick point on the trajectory corresponds to the third factor.*

Matching of neurons along the second trajectory in Fig. 4 with the list of deputies has shown that they correspond to the members of the fraction "United Russia" (UR). This fraction is the largest one, consist totally of 285 deputies, but it is less homogeneous. Therefore the Lyapunov function along the trajectory is relatively low and it has no turning point.

Fig. 5 shows trajectories of neurodynamics after additional deleting the third factor from the network. Two remaining trajectories contain members of UR and independent deputies (ID). The upper trajectory contains only members of UR and lower one – mainly ID but also members of UR. This is an additional evidence of UR heterogeneity. Factors UR and ID were identified by minimums of the second derivatives along the corresponding trajectories. The general relation between the parliament fractions and obtained factors is shown in Table 1.

Tab. 1. Relation between parliament fractions and factors

	1	2	3	4	5
CPRF	0 / 0	51/49	0 / 0	0 / 2	0 / 0
LDPR	1 / 2	0 / 0	36/ 35	0 / 0	0 / 0
ML	3 / 3	0 / 0	0 / 0	37/ 38	1 / 0
ID	1 / 14	0 / 0	0 / 1	0 / 1	15 / 0

The fit between the fractions and the factors was evaluated by F-measure [9]. Averaged over all fractions it amounted to 0.98.

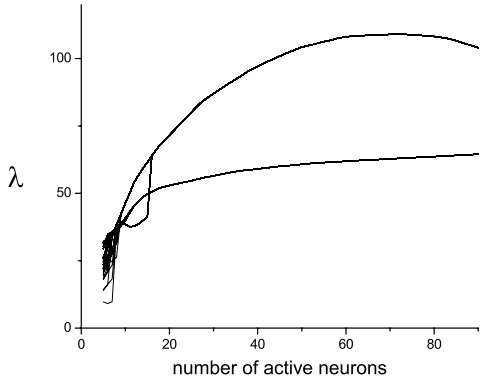

Fig. 5 *The same as in Figs. 3 and 4 after deleting three first factors.*

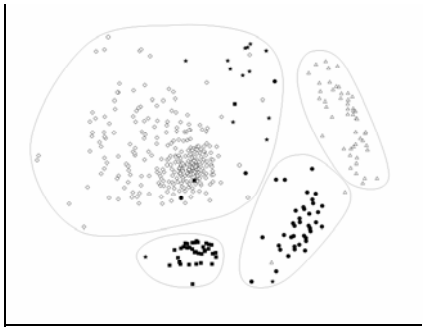

Fig. 6 *Two-dimensional map of voting parliament members. Thin lines - borders of clusters. ♦ - UR, □ – CPRF, ■ – LDPR, ● - ML, ★ – ID.*

We compared our results with those obtained using some traditional clustering methods [8]. First, we clustered the parliament members with the direct use of a similarity matrix. Similarity between two

deputies was calculated by comparison of vectors of their voting. We used different measures of similarity: Euclidian distance, cosine, Jaccard and Dice. Both hierarchial and *k*-means clustering gave clusters far from parliament fractions: all fractions intersected in clusters and fraction LDPR could not be separated from ER at all.
Second, we performed mapping of parliament members by the method of multidimensional scaling. The results are shown in Fig. 6. This map was clustered. The borders of clusters are shown by thin lines. Generally, as factors obtained before, clusters coincide with parliament fractions except for independent deputies. The results of clustering and factorization are compared in the Table. The mean F-measure amounted to 0.95 that is slightly smaller than that obtained for factors.

4. Conclusion

The Hopfield-like neural network is capable of performing Boolean factor analysis of the signals of high dimension and complexity. We described the new method of sequential factor extraction based on the Lyapunov function value and change during the neural network active dynamics. This method is based on the two phenomena discovered by means of the neural network behavior analysis using the simulated data. Then we demonstrated procedure effective application on the real data. In our previous papers we showed its high efficiency in case of textual data analysis. Here its ability is demonstrated in the field of politics.

References:

[1] J. De Leeuw, Principal Component Analysis of Binary data, Application to Roll-Call Analysis, 2003, http://gifi.stat.ucla.edu.

[2] A. A. Frolov, D. Husek and I. P. Muraviev, “Informational Efficiency of Sparsely Encoded Hopfield-like Autoassociative Memory”, Optical Memory & Neural Networks, Vol. 12, 2003, pp. 177 - 197.

[3] A. A. Frolov, A. M. Sirota, D. Husek, I. P. Muraviev and P. Y. Polyakov, “Binary Factorization in Hopfield-like Neural Networks: Single-Step Approximation and Computer Simulations”, Neural Networks World, Vol. 14, 2004, pp. 139 - 152.

[4] A. A. Frolov, D. Husek, P. Y. Polyakov, H. Rezankova and V. Snasel, “Binary Factorization of Textual Data by Hopfield-like Neural Network”. In: COMPSTAT 2004, Physica-Verlag, Heidelberg, 2004, pp. 1027 - 1034.

[5] D. Husek, A. A. Frolov, H. Rezankova, V. Snasel and P. Y. Polyakov, “Neural Network Nonlinear Factor Analysis of High Dimensional Binary Signals”. In: SITIS 2005, University of Bourgogne, Dijon, 2005, pp. 86 - 89.

[6] P. Y. Polyakov, A. A. Frolov and D. Husek, “Binary Factor Analysis by Hopfiled Network and its Application to Automatic Text Classification”. In: Neuroinformatics 2006, MIFI, Moscow, Russia, 2006.

[7] http://www.indem.ru/indemstat.

[8] http://www.publicwhip.org.uk.

[9]http://www.dcs.gla.ac.uk/Keith/Chapter.7/Ch.7.html.

Acknowledgement: This work was partially supported by grant RFBR No. 05-07-90049, by the projects No. 1ET100300414, 201/05/0079 and by the Institutional Research Plan AVOZ10300504 awarded by the Grant Agency of the Czech Republic.