Neural Network Boolean Factor Analysis and Application

DUSAN HUSEK
ICS AS CR, v.v.i.
Pod Vodárenskou
věží 2,
182 07 Prague 8
Czech Republic

ALEXANDER FROLOV
Institute of Higher
Nervous Activity
and Neurophysiology
Russian Academy of Sciences
Butlerova 5a,
117 485 Moscow
Russia

PAVEL POLYAKOV
Institute of Optical
Neural Technologies
Russian AS
Vavilova 44,
119 333 Moscow
Russia

VACLAV SNASEL
VSB TU Ostrava
Departmet of
Computer Science
17 listopadu 15,
70 833 Ostrava
Czech Republic

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 roll-call voting problem. The method of sequential factor extraction, based on the Lyapunov function is discussed in deep. Efficiency of this attempt is shown on simulated data and on real data from Russian parliament as well.

Key–Words: Hopfield Neural Network, Boolean Factor Analysis,Unsupervised Learning, Dimension Reduction, Data Mining

1 Introduction

Theoretical analysis and computer simulations performed in (5) revealed that Hopfield-like neural networks are capable of performing boolean factor analysis signals of high dimension and complexity. Factor analysis is a procedure which maps original signals into the space of factors. The principal component (PC) analysis is a classical example of such mapping in the linear case. Linear factor analysis implies that each original case can be presented as

$$\mathbf{X} = \mathbf{F}\mathbf{S} + \varepsilon \tag{1}$$

where $\mathbf{F}$ is a matrix $N \times L$ of factor loadings, $\mathbf{S}$ is a vector of factor scores and ε an error. Each component of $\mathbf{S}$ gives contribution of corresponding factor in the original signal. Columns of loading matrix $\mathbf{F}$ give vectors presenting corresponding factors in the signal space. Below namely these vectors are termed factors. Mapping of original space to the factor space means that signals are represented by vectors $\mathbf{S}$ instead of original vectors $\mathbf{X}$. Dimensionality of vectors $\mathbf{S}$ is much less than dimensionality of signals $\mathbf{X}$. Thereby 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:

$$\mathbf{X} = \bigvee S_l \mathbf{f}^l. \tag{2}$$

In this case, original signals, factor scores and factor loadings are binary and mapping of original signal to the factor space means citation of factors that were mixed in the signal. The mean number of factors mixed in the signals we term signal complexity C.

For the case of large dimensionality and complexity of signals it was a challenge (5) to utilize for Boolean factor analysis the Hopfield-like neural network with parallel dynamics. Binary patterns $\mathbf{X}$ 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 $\mathbf{X}^m$ are stored in the matrix of synaptic connections $\mathbf{J}'$ according to correlational Hebbian rule:

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

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 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 Hebbian rule:

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

where $q_i = \sum_{m=1}^{M} X_i^m/M$ is a mean activity of 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 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 attractor of network dynamics. This property of factors is a base of the used two-run procedure of factor search. Its initialization starts by presentation of random initial pattern $\mathbf{X}^{in}$ with $k_{in} = r_{in}N$ active neurons. Activity k_{in} is supposed to be much smaller than activity of factors. On presentation of $\mathbf{X}^{in}$, network activity $\mathbf{X}$ evolves to some attractor. The evolution is determined by the parallel dynamics equation in discrete time. At each time step:

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

where h_i are components of the vector of synaptic excitations

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

Θ is step function, and $T(t)$ is activation threshold. The first term in (6) gives synaptic excitations provided by the principal neurons of 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 J'_{ij}. Thus (6) can be rewritten as $h_i(t) = \sum_{j=1}^{N} J_{ij}X_j(t)$ where $\mathbf{J} = \mathbf{J}' - \mathbf{M}\mathbf{q}\mathbf{q}^{\mathbf{T}}$, $\mathbf{q}$ is a vector with components $q_i - q$ and $\mathbf{q}^T$ is a transposed $\mathbf{q}$. As shown in (8) the replacement of 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 level of the activation threshold, small random noise was added to the activation threshold 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 be added to the neurons’ thresholds. Noise to individual neurons was fixed during the whole recall process to provide its convergence. As shown in (5), this choice of activation thresholds allows for stabilization of the network activity in point or cyclic attractor of length two.

When 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 network activity evolves to some attractor at the new level of activity $k_{in}+1$. Then level of activity increases to $k_{in}+2$, and so on, until number of active neurons reaches the final level $r_f N$ with $r_f > p$. 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 fixed level of activity.

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

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

where $\mathbf{X}^T(t+1)$ and $\mathbf{X}(t)$ are two network states in cyclic attractor (for point attractor $\mathbf{X}^T(t+1) = \mathbf{X}(t)$). 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 the most trials of the recall process. Thus, more and more trials are required to obtain new attractor with relatively small value of Lyapunov function. To overcome this problem the dominant attractors should be deleted from the network memory. The deletion was performed according to Hebbian unlearning rule by substraction $\Delta J_{ij}, j \neq i$ from synaptic connections J_{ij} where

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

$$+(X_j(t-1)-r)(X_i(t)-r), \qquad (8)$$

$J(\mathbf{X})$ is the average synaptic connection between active neurons of the attractor, $\mathbf{X}(t-1)$ and $\mathbf{X}(t)$ are patterns of network activity at last time steps of iteration process, r is the level of activity, and η is an unlearning rate. For point attractor $\mathbf{X}(t) = \mathbf{X}(t-1)$ and for cyclic attractor $\mathbf{X}(t-1)$ and $\mathbf{X}(t)$ are two states of attractor.

There are three important similarities between the described procedure of Boolean factor analysis and linear PC analysis. First, PC analysis is based on the same covariation matrix as connection matrix in Hopfield network. Second, factor search in PC analysis can be performed by the iteration procedure similar to that described by (5) and (6) but binarization of synaptic excitations by the step function must be replaced by their normalization:$X_i(t+1) = h_i(t)/|\mathbf{h}(t)|$. Then iteration procedure starting from any random state converges to the eigenvector $\mathbf{f}_1$ of covariation matrix with the largest eigenvalue Λ_1. Just this eigenvector is treated as the first factor in PC analysis. 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, Boolean factor analysis by Hopfield-like network has one principal difference from linear PC analysis. Attractors of iteration procedure in PC analysis 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 in view of Boolean factor analysis by Hopfield-like network. First, how often would network activity converge to one of the factors starting from 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 Boolean factor analysis is required (1). In our previous papers (6),(7) and (8) we performed it for analysis of textual data. Here we apply our method to the data for Russian parliament voting. The results are described in Section 3.

1.1 Artificial Signals

To reveal peculiarities of true and spurious attractors we performed computer experiments with artificial signals. Each pattern of the learning set is supposed to be a Boolean superposition of exactly C factors and each factor is supposed to contain exactly $n = pN$ 1-s and $(1-p)N$ 0-s. Thus, each factor $\mathbf{f}^l \in B_n^N$ and for each pattern of the learning set, vector of factor scores $\mathbf{S} \in B_C^L$ where $B_n^N = \left\{ \mathbf{X} | X_i \in \{0,1\}, \sum_{i=1}^{N} X_i = n \right\}$. We supposed factor loadings and factor scores to be statistically independent. As an example, Fig. 1 demonstrates changes of relative Lyapunov function for $N = 3000$, $L = 5300$, $p = 0.02$ and $C = 10$. Recall process started at $r_{in} = 0.005$.

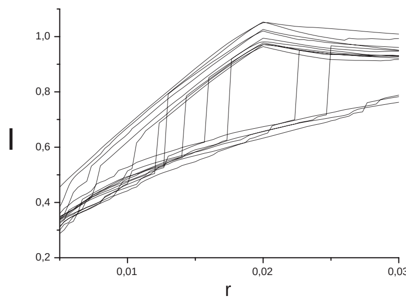

Fig. 1 *Fig.1. Relative Lyapunov function λ in dependence on the relative network activity r. The data were normalized to the mean of Lyapunov function for true attractors at $r = p$.*

Trajectories of network dynamics form two separated groups. As shown in Fig. 2, the trajectories with higher values of Lyapunov function are true and with lower ones are spurious. This Figure relates values of Lyapunov function for patterns of network activity at points $r = p$ to maximal overlaps of these patterns with factors. Overlap between two patterns $\mathbf{X}^1$ and $\mathbf{X}^2$ with p active neurons was calculated by formula

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

By this formula overlap between equal patterns is equal to 1 and mean overlap between independent patterns is equal to 0. Patterns with high Lyapunov function have high overlap with one of the factors, while patterns with 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 Lyapunov function are normalized by mean value of this function over true attractors at the point $r = p$.

The second characteristic feature of true trajectories is the existence of a kink at a point $r = p$ where the level of network activity coincides with that in factors (see Fig. 1). When $r < p$ the increase of r results in almost linear increase of the relative Lyapunov function. Increase of r occurs in this case due to joining of neurons belonging to the factor that are strongly

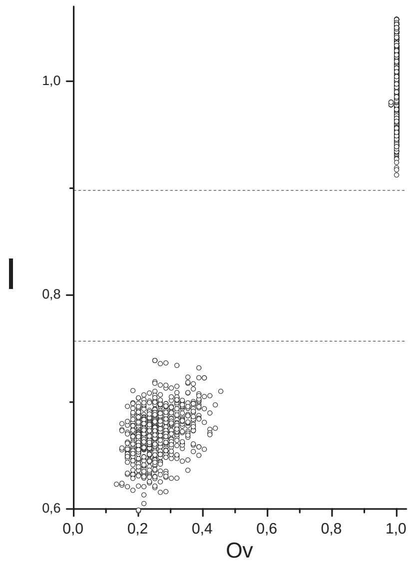

Fig. 2 *Values of normalized Lyapunov function in relation to overlaps with the closest factors. It is shown that true and spurious attractors can be easily separated due to a large gap between distributions of values of Lyapunov function.*

connected with other neurons of factor. Then joining of new neurons results in proportional increase of mean synaptic excitation to the active neurons of 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 factor by week connections. Thus, the increase of the relative Lyapunov function for true trajectory sharply slows and it tends to the values of Lyapunov function for spurious trajectories.

The third characteristic feature of true trajectories which allows for their distinction from spurious trajectories, lies in the behavior of their activation thresholds. Activation thresholds T at the final states of external steps are shown in Fig. 3 for the same network parameters as in Fig. 1.

Initially activation thresholds for true attractors are larger in comparison with spurious ones. However at the point $r = p$ they sharply drop to the level of spurious attractors. This behavior is originated from the well known fact (see, for example, (8)) that during activation of the fragment of one of the stored patterns the distribution of synaptic excitations has two separated high and low modes: high - for neurons belonging to the pattern and low - for neurons not belonging to it. The mean of the high mode increases proportionally to the size of the fragment and the mean of the low mode is close to zero. When $r < p$ fragments of stored patterns are activated along the true trajectories. Then activation thresholds are inside the high mode.

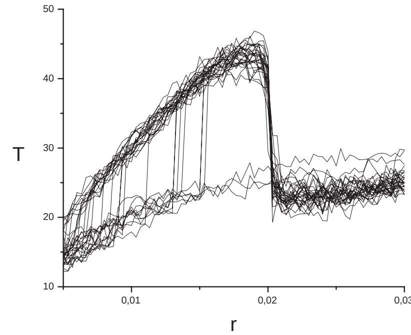

Fig. 3 *Fig.3. The change of activation threshold T along the recall process. Notice the drop of activation threshold at the point $r = p$. $r = p$.*

The mean of high mode increases when r increases since r is the relative size of activated fragment of the stored pattern. Consequently the activation thresholds increase with r. However when the stored pattern is activated totally ($r = p$), the activation threshold has to jump to the low mode to activate additional neurons not belonging to the pattern as r continues to increase. The use of these three features of true trajectories provides reliable recognition of factors. This statement was clearly confirmed in our previous papers (6), (7) and (8) where our method was performed for analysis of textual data. Here we apply it for analysis of parliament voting.

2 Analysis of Parliament Voting

The analysis was performed for results of roll-call votes in Russian parliament in 2004 (9). Each vote is considered as binary vector with component 1 if the correspondent deputy voted affirmatively and 0 negatively. The number of deputies (consequently the dimensionality of signal space and network size) amounts 450.

The number of votes during the year amounts 3150. Fig. 4 shows Lyapunov function along trajectories starting from 1500 random initial states. All these states converge to four trajectories. Two of them have obvious kinks and therefore were identified as two factors. The factor with highest Lyapunov function contains 50 deputies and completely coincides with the fraction of Communist Party (CPRF). Another factor contains 36 deputies. All of them belong to fraction of Liberal-Democratic Party (LDPR) which contains totally 37 deputes. Thus one of the members of this fraction fell out of the corresponding factor. Pointed kinks at the corresponding trajectories give evidence that these fractions are the most discipline and their members vote coherently.

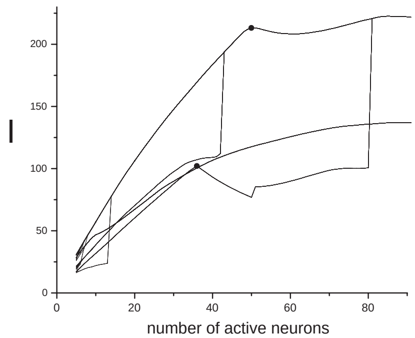

Fig. 4 *Relative Lyapunov function λ for parliament data in dependence on the number of active neurons. Thick points are klinks of the first and second factors.*

Fig. 5 demonstrates trajectories after deleting of the two factors. Starting from 1500 initial states they converge to only two trajectories. One of them has a kink but it is not so strict as for CPRF and LDPR factors. We supposed that the point where the second derivative of Lyapunov function by k has maximum by absolute value is the third factor. The factor contains 37 deputies. All of them belong to the fraction "Rodina" which contains totally 40 deputes. Thus 3 of its members fell out of the factor. The fuzziness of kink at the trajectory gives evidence that this fraction is not so homogeneous as two first ones and actually the fraction split at two fractions in 2005.

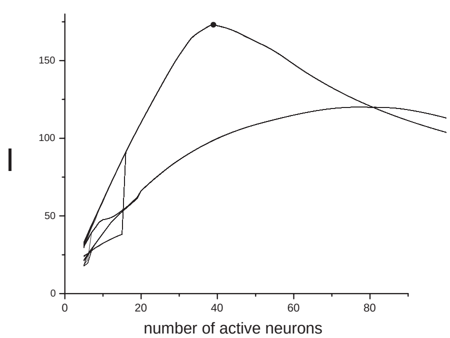

Fig. 5 *The same as in Fig. 4 after deleting two first factors. Thick point is klink of third factor.*

Matching of neurons along the second trajectory in Fig. 5 with the list of deputes showed that it corresponds to the fraction "United Russia" (UR). This fraction is the largest and contains totally 280 deputes but it is less homogeneous. Therefore Lyapunov function along the trajectory is low and it has no kink at all.

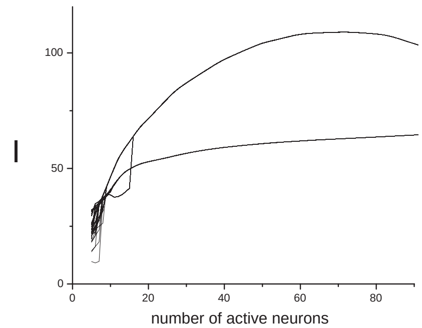

Fig. 6 *The same as in Figs. 4 and 5 after deleting three first factors.*

Fig. 6 shows trajectories of neurodynamics after additional deleting the third factor from the network. The remaining two trajectories contain only members of UR. The states of different trajectories do not intersect, so the UR fraction is evidently divided to two subfractions.

3 Conclusion

Hopfield-like neural network is capable of performing Boolean factor analysis of the signals of high dimension and complexity. In this case analysis of a voting. But it is suitable for much complicated task analysis. In our previous papers we showed its high efficiency in analysis of textual data. Here its ability is demonstrated in the field of politics. More detailed here used method description can be found in work (2; 3; 4). The idea of the method is to exploit the well known property of the network to create attractors of the network dynamics by the tightly connected groups of neurons. We supposed, first, that each topic is characterized by the set of specific words (often called concepts) which appear in the relevant documents coherently and, second, that different concepts are presented in documents in random combinations.

Acknowledgements: The research was funded by the Center of Applied Cybernetics (grant No. 1M6840070004) and by the Institutional Research Plan AV0Z10300504 Computer Science for the Information Society: Models, Algorithms, Applications, by the National Research Program of the Czech Republic (project No. 1ET100300414) and by of the Grant Agency of the Czech Republic (Grant No. 201/05/0079).

References:

[1] De Leeuw J. (2003) *Principal component analysis of binary data. Application to roll-call analysis.* http://gifi.stat.ucla.edu

[2] Frolov AA, Husek D, Muraviev IP, Polyakov PY. 2007. Boolean factor analysis by attractor neural network. IEEE Tran on Neural Networks, **18**,3:698-707.

[3] Frolov AA, Husek D, Polyakov PY. 2008. Recurrent neural network based Boolean factor analysis procedure and its application to automatic papers categorization, IEEE Tran on Neural Networks, Submited 2007

[4] Frolov AA, Husek D, Polyakov PY. 2008. Origin and elimination of two global spurious attractors in Hopfield-like network performing Boolean factor analysis. JMLR, Submited 2007

[5] Frolov A.A., Sirota A.M., Husek D., Muraviev I.P., Polyakov P.J. (2004). *Binary factorization in Hopfield-like neural networks: single-step approximation and computer simulations.* Neural Networks World, **14**, 139-152.

[6] Frolov A.A., Husek Dusan, Polyakov P.A., Rezankova H., Snasel Vaclav (2004) *Binary Factorization of Textual Data by Hopfield-Like Neural Network.* In: Computational Statistics. (Ed.: Antoch J.) - Heidelberg, Physica Verlag 2004, pp. 1035-1041 (ISBN:3-7908-1554-3) Held: COMPSTAT 2004. Symposium /16./, Prague, CZ, 04.08.23-04.08.27

[7] Husek D., Frolov AA., Rezankova, H., Snasel, V., Polyakov, P. (2005) *Neural Network Nonlinear Factor Analysis of High Dimensional Binary Signals.* Yaound 27.11.2005 - 1.12.2005. In: SITIS 2005. Dijon : University of Bourgogne, 2005, s. 86-89. ISBN 2-9525435-0.

[8] Polyakov P.Y., Frolov A.A., Husek D. (2006) *Binary factor analysis by Hopfiled network and its application to automatic text classification.* Neuroinformatics-2006, Russia.

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