Comparison of Neural Network Boolean Factor Analysis Method with Some Other Dimension Reduction Methods on Bars Problem^{*}

Dušan Húsek¹, Pavel Moravec², Václav Snášel², Alexander Frolov³,
Hana Řezanková⁴, and Pavel Polyakov⁵

¹ Institute of Computer Science, Dept. of Neural Networks,
Academy of Sciences of Czech Republic, Pod Vodárenskou věží 2,
182 07 Prague, Czech Republic

`dušan@cs.cas.cz`

² Department of Computer Science, FEECS, VŠB – Technical University of Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
{`pavel.moravec, vaclav.snasel`}@vsb.cz

³ Institute of Higher Nervous Activity and Neurophysiology, Russian Academy of Sciences,
Butlerova 5a, 117 485 Moscow, Russia

`aafrolov@mail.ru`,

⁴ Department of Statistics and Probability, University of Economics,
Prague, W. Churchill sq. 4, 130 67 Prague, Czech Republic

`rezanka@vse.cz`

⁵ Institute of Optical Neural Technologies, Russian Academy of Sciences,
Vavilova 44, 119 333 Moscow, Russia

`pavel.mipt@mail.ru`

Abstract. In this paper, we compare performance of novel neural network based algorithm for Boolean factor analysis with several dimension reduction techniques as a tool for feature extraction. Compared are namely singular value decomposition, semi-discrete decomposition and non-negative matrix factorization algorithms, including some cluster analysis methods as well. Even if the mainly mentioned methods are linear, it is interesting to compare them with neural network based Boolean factor analysis, because they are well elaborated. Second reason for this is to show basic differences between Boolean and linear case. So called bars problem is used as the benchmark. Set of artificial signals generated as a Boolean sum of given number of bars is analyzed by these methods. Resulting images show that Boolean factor analysis is upmost suitable method for this kind of data.

1 Introduction

In order to perform object recognition (no matter which one) it is necessary to learn representations of the underlying characteristic components. Such components correspond

^{*} The work was partly funded by the Centre of Applied Cybernetics 1M6840070004 and partly by the Institutional Research Plan AV0Z10300504 "Computer Science for the Information Society: Models, Algorithms, Applications" and by the project 1ET100300419 of the Program Information Society of the Thematic Program II of the National Research Program of the Czech Republic.

to objects, object-parts, or features. An image usually contains a number of different objects, parts, or features and these components can occur in different configurations to form many distinct images. Identifying the underlying components which are combined to form images is thus essential for learning the perceptual representations necessary for performing object recognition.

There exists many attempts that could be used for this reason. In this paper we concentrate on *neural network based algorithm for Boolean factor analysis* and compare it with some recently used dimension reduction techniques for automatic feature extraction. Here we empirically investigate their performance.

How does the brain form a useful representation of its environment? That is question that was behind the development of neural network based methods for dimension reduction. Among those neural network based Boolean factor analysis, see Frolov at al. [1,2], and Földiáks network [3] fall.

The comparison of linear approaches [4,5,6,7,8] capable to find an approximate solution of the Boolean factor analysis task with our attempt [2], if only for the reason they are fast and well elaborated, is very interesting. The most popular linear method is the *singular value decomposition* which was already successfully used for automatic feature extraction. However singular value decomposition is not suitable for huge data collections and is computationally expensive, so other methods of dimension reduction were proposed. Here we apply *semi-discrete decomposition* as example. As the data matrix has all elements non-negative, a new method called *non-negative matrix factorization* is applied as well.

However, for the sake of finding an hidden structure in the images data statistical methods used very often in the past are still good choice, mainly different algorithms for cluster analysis, perhaps in conjunction with a some data visualization methods.

The bars problem (and its variations) is a benchmark task for the learning of independent image features [3]. The collection, the *base vectors* can be interpreted as images, describing some common characteristics of several input signals. These base vectors are often called eigenfaces in the case of face recognition task, see M. Turk and A. Pentland [9].

The rest of this paper is organized as follows. The second section explains the dimension reduction method used in this study. Then in the section three we describe experimental results, and finally in the section four we made some conclusions.

2 Dimension Reduction

We used the three promising linear methods of dimension reduction for our comparison with Boolean factor analysis – singular value decomposition, semi-discrete decomposition, non-negative matrix factorization and two statistical clustering methods. All of them are briefly described bellow.

2.1 Neural Network Boolean Factor Analysis

Neural network Boolean Factor Analysis (*NBFA*) is a powerful method for revealing the information redundancy of high dimensional binary signals [2]. It allows to express

every signal (vector of variables) from binary data matrix A of observations as superposition of binary factors:

$$A = \bigvee_{l=1}^L S_l f^l, \quad (1)$$

where S_l is a component of factor scores and f^l is a vector of factor loadings and $\vee$ denotes Boolean summation ($0 \vee 0 = 0, 1 \vee 0 = 1, 0 \vee 1 = 1, 1 \vee 1 = 1$). If we mark Boolean matrix multiplication by the symbol $\odot$, then we can express approximation of data matrix A in matrix notation

$$A_k \simeq F \odot S \quad (2)$$

where S is the matrix of factor scores and F is the matrix of factor loadings. The Boolean factor analysis implies that components of original signals, factor loadings and factor scores are binary values.

Optimal solution of A_k decomposition according to (2) by brute force search is NP-hard problem and as such is not suitable for high dimensional data. On other side the classical linear methods could not take into account non-linearity of Boolean summation and therefore are inadequate for this task.

2.2 Singular Value Decomposition

Singular Value Decomposition (SVD), see M. Berry, S. Dumais, and T. Letsche [5], is an algebraic extension of classical vector model. It is similar to the PCA method, which was originally used for the generation of eigenfaces. Informally, SVD discovers significant properties and represents the images as linear combinations of the base vectors. Moreover, the base vectors are ordered according to their significance for the reconstructed image, which allows us to consider only the first k base vectors as important (the remaining ones are interpreted as "noise" and discarded). Furthermore, SVD is often referred to as more successful in recall when compared to querying whole image vectors (M. Berry, S. Dumais, and T. Letsche [5]).

Formally, we decompose the matrix of images A by singular value decomposition, calculating singular values and singular vectors of A .

We have matrix A , which is an $n \times m$ rank- r matrix (where $m \geq n$ without loss of generality) and values $\sigma_1, \dots, \sigma_r$ are calculated from eigenvalues of matrix AA^T as $\sigma_i = \sqrt{\lambda_i}$. Based on them, we can calculate column-orthonormal matrices $U = (u_1, \dots, u_n)$ and $V = (v_1, \dots, v_m)$, where $U^T U = I_n$ and $V^T V = I_m$, and a diagonal matrix $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$, where $\sigma_i > 0$ for $i \leq r$, $\sigma_i \geq \sigma_{i+1}$ and $\sigma_{r+1} = \dots = \sigma_n = 0$.

The decomposition

$$A = U \Sigma V^T \quad (3)$$

is called *singular decomposition* of matrix A and the numbers $\sigma_1, \dots, \sigma_r$ are *singular values* of the matrix A . Columns of U (or V) are called *left* (or *right*) singular vectors of matrix A .

Now we have a decomposition of the original matrix of images A . We get r nonzero singular numbers, where r is the rank of the original matrix A . Because the singular

values usually fall quickly, we can take only k greatest singular values with the corresponding singular vector coordinates and create a k -reduced singular decomposition of A .

Let us have k ($0 < k < r$) and singular value decomposition of A

$$A = U\Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix} \quad (4)$$

We call $A_k = U_k \Sigma_k V_k^T$ a k -reduced singular value decomposition (rank- k SVD).

Instead of the A_k matrix, a matrix of image vectors in reduced space $D_k = \Sigma_k V_k^T$ is used in SVD as the representation of image collection. The image vectors (columns in D_k) are now represented as points in k -dimensional space (the *feature-space*). represent the matrices U_k, Σ_k, V_k^T .

Rank- k SVD is the best rank- k approximation of the original matrix A . This means that any other decomposition will increase the approximation error, calculated as a sum of squares (*Frobenius norm*) of error matrix $B = A - A_k$. However, it does not implicate that we could not obtain better precision and recall values with a different approximation.

Once computed, SVD reflects only the decomposition of original matrix of images. If several hundreds of images have to be added to existing decomposition (*folding-in*), the decomposition may become inaccurate. Because the recalculation of SVD is expensive, so it is impossible to recalculate SVD every time images are inserted. The *SVD-Updating* is a partial solution, but since the error slightly increases with inserted images. If the updates happen frequently, the recalculation of SVD may be needed soon or later.

2.3 Semi-discrete Decomposition

Semi-Discrete Decomposition (*SDD*) is one of other SVD based methods, proposed recently for text retrieval in (T. G. Kolda and D. P. O'Leary [8]). As mentioned earlier, the rank- k SVD method (called *truncated SVD* by authors of semi-discrete decomposition) produces dense matrices U and V , so the resulting required storage may be even larger than the one needed by the original term-by-document matrix A .

To improve the required storage size and query time, the semi-discrete decomposition was defined as

$$A \approx A_k = X_k D_k Y_k^T, \quad (5)$$

where each coordinate of X_k and Y_k is constrained to have entries from the set $\varphi = \{-1, 0, 1\}$, and the matrix D_k is a diagonal matrix with positive coordinates.

The SDD does not reproduce A exactly, even if $k = n$, but it uses very little storage with respect to the observed accuracy of the approximation. A rank- k SDD (although from mathematical standpoint it is a sum on rank-1 matrices) requires the storage of $k(m + n)$ values from the set $\{-1, 0, 1\}$ and k scalars. The scalars need to be only single precision because the algorithm is self-correcting. The SDD approximation is formed iteratively.

The optimal choice of the triplets (x_i, d_i, y_i) for given k can be determined using greedy algorithm, based on the residual $R_k = A - A_{k-1}$ (where A_0 is a zero matrix).

2.4 Non-negative Matrix Factorization

The Non-negative Matrix Factorization (*NMF*) method calculates an approximation of the matrix A as a product of two matrices, W and H . The matrices are usually pre-filled with random values (or H is initialized to zero and W is randomly generated). During the calculation the values in W and H stay positive. The approximation of matrix A , matrix A_k , can be calculated as $A_k = WH$.

The original NMF method tries to minimize the Frobenius norm of the difference between A and A'_k using

$$\min_{W, H} ||V - WH||_F^2 \quad (6)$$

as the criterion in the minimization problem.

Recently, a new method was proposed in (M. W. Spratling [10]), where the constrained least squares problem is solved

$$\min_{H_j} \{ ||V_j - WH_j||_2^2 - \lambda ||H_j||_2^2 \} \quad (7)$$

as the criterion in the minimization problem. This approach gives yields better results for sparse matrices.

Unlike in SVD, the base vectors are not ordered from the most general one and we have to calculate the decomposition for each value of k separately.

2.5 Statistical Clustering Methods

To set our method in more global context we applied cluster analysis, too. The clustering methods help to reveal groups of similar features, it means typical parts of images. However, obtaining disjunctive clusters is a problem of the usage of traditional hard clustering. We have chosen two most promising techniques available in recent statistical packages – hierarchical agglomerative algorithm and two-step cluster analysis.

A hierarchical agglomerative algorithm (*HAA*) starts with each feature in a group of its own. Then it merges clusters until only one large cluster remains which includes all features. The user must choose dissimilarity or similarity measure and agglomerative procedure. At the first step, when each feature represents its own cluster, the dissimilarity between two features is defined by the chosen dissimilarity measure. However, once several features have been linked together, we need a linkage or amalgamation rule to determine when two clusters are sufficiently similar to be linked together. Several linkage rules have been proposed. For example, the distance between two different clusters can be determined by the greatest distance between two features in the clusters (complete linkage method – *CL*), or average distance between all pairs of objects in the two clusters (average linkage between groups – *ALBG*). Hierarchical clustering is based on the proximity matrix (dissimilarities for all pairs of features) and it is independent on the order of features.

For binary data (one for the black point and zero for the white point), we can choose for example Jaccard and Ochiai (cosine) similarity measures of two features. The former can be expressed as $S_J = \frac{a}{a+b+c}$ where a is the number of the common occurrences of ones and $b+c$ is the number of pairs in which one value is one and the second is zero.

The latter can be expressed as $S_O = \sqrt{\frac{a}{a+b} \cdot \frac{a}{a+c}}$. The further clustering techniques mentioned above are suitable for large data files but they are independent on the order of features.

In two-step cluster analysis (TSCA), the features are arranged into sub-clusters, known as cluster features (CF), first. These cluster features are then clustered into k groups, using a traditional hierarchical clustering procedure. A cluster feature represents a set of summary statistics on a subset of the data. The algorithm consists of two phases. In the first one, an initial CF tree is built (a multi-level compression of the data that tries to preserve the inherent clustering structure of the data). In the second one, an arbitrary clustering algorithm is used to cluster the leaf nodes of the CF tree. Advantage of this method is its ability to work with larger data sets; disadvantage then, is its sensitivity to the order of the objects (features in our case). In the implementation in the SPSS system, the log-likelihood distance is proposed.

This distance between clusters a and b is

$$d(a, b) = \zeta_a + \zeta_b - \zeta_{<a,b>}$$

where $<a, b>$ denotes a cluster created by joining objects from clusters a and b , and

$$\xi_g = -m_g \sum_{i=1}^n H_{gi} ,$$

where m_g is the number of features in the g^{th} cluster, n is the number of images and H_{gi} is an entropy

$$H_{gi} = - \sum_{u=1}^2 \frac{m_{giu}}{m_g} \ln \frac{m_{giu}}{m_g} ,$$

where m_{gi1} is the number of zeros and m_{gi2} is the number of ones.

3 Experimental Results

For testing of above mentioned methods, we used generic collection of 1600 32×32 black-and-white images containing different combinations of horizontal and vertical lines (bars). The probabilities of bars to occur in images were the same and equal to $10/64$, i.e. images contain 10 bars in average. An example of several images from generated collection is shown in Figure 1a.

For the first view on image structures, we applied traditional cluster analysis. We clustered 1024 (32×32) positions into 64 and 32 clusters. The problem of the use of traditional cluster analysis consists in that we obtain disjunctive clusters. So we can find only parts of horizontal or vertical bars respectively.

We applied HAA and TSCA algorithms in the SPSS system. The problem of the latter consists in that it is dependent on the order of features. We used two different orders; in the second one the images from 1001 to 1009 were placed between the images 100 and 101 for example.

Fig. 1. Several bars from generated collection (a) First 64 base images of bars for SVD method (b) First 64 factors for original NMF method (c)

For hierarchical cluster analysis, we tried to use different similarity measures. We found that linkage methods have more influence to the results of clustering than similarity measures. We used Jaccard and Ochiai (cosine) similarity measures suitable for asymmetric binary attributes. We found both as suitable methods for the identification of the bars or their parts. For 64 clusters, the differences were only in a few assignments of positions by ALBG and CL methods with Jaccard and Ochiai measures.

The figures illustrate the application of some of these techniques for 64 and 32 clusters. Figure 2a,b show results of ALBG method with Jaccard measure. Figure 2a for 32 clusters and Figure 2b for 64 clusters. In the case of 32 clusters, we found 32 horizontal bars (see Figure 2c) by TSCA method for the second order of features.

Many of tested methods were able to generate a set of base images or factors, which should ideally record all possible bar positions. However, not all methods were truly successful in this.

With SVD, we obtain classic singular vectors, the most general being among the first. The first few are shown in Figure 1b. We can see, that the bars are not separated and different shades of gray appear.

The NMF methods yield different results. The original NMF method, based on the adjustment of random matrices W and H provides hardly-recognizable images even for

Fig. 2. ALBG Jaccard – 32 clusters (a) ALBG Jaccard – 64 clusters (b) TSCA – 32 clusters (c)

Fig. 3. First 64 factors for GD-CLS NMF method (a) First 64 base vectors for SDD method (b) First 64 base vectors for NBFA method (c)

$k = 100$ and 1000 iterations (we used 100 iterations for other experiments). Moreover, these base images still contain significant salt and pepper noise and have a bad contrast. The factors are shown in Figure 1c. We must also note, that the NMF decomposition will yield slightly different results each time it is run, because the matrix(es) are pre-filled with random values.

The GD-CLS modification of NMF method (proposed in [7]) tries to improve the decomposition by calculating the constrained least squares problem. This leads to a better overall quality, however, the decomposition really depends on the pre-filled random matrix H . The result is shown in Figure 3a.

The SDD method differs slightly from previous methods, since each factor contains only values $\{-1, 0, 1\}$. Gray in the factors shown in Figure 3b represents 0; -1 and 1 are represented with black and white respectively.

The base vectors in Figure 3b can be divided into three categories:

1. Base vectors containing only one bar.
2. Base vectors containing one horizontal and one vertical bar.
3. Other base vectors, containing several bars and in some cases even noise.

Finally, we made decomposition of images into binary vectors by Boolean factor analysis method proposed in [11]. The factor search was performed under assumption that the number of ones in factor is not less than 5 and not greater than 200. Since the images are obtained by Boolean summation of binary bars, it is not surprising, that NBFA is able to reconstruct all bars as base vectors, providing an ideal solution, as we can see in Figure 3c.

4 Conclusion

In this paper, we have compared several dimension reduction and clustering methods on bars collection. NBFA perfectly found basis (factors) from which the all learning pictures can be reconstructed. It is because factor analysis is looking for hidden dependencies among observed data. On the other side reason for 100% success resides also in this, that there was no noise superposed on the input data.

Cluster analysis is focused on finding original factors from which images were generated. Applied clustering methods were quite successful in finding these factors. The

problem consists in created disjunctive clusters. So, only some bars or their parts were revealed. However, from the general view on 64 clusters, it is obvious, that images are compounded from vertical and horizontal bars (lines). By two-step cluster analysis 32 horizontal lines were revealed by clustering to 32 clusters.

The SVD method may also be used, but the results are not as good as in the case of NBFA. The SVD and NMF methods yield slightly worse results, since they are not focused on binary data.

References

1. Frolov, A.A., Sirota, A.M., Húsek, D., Muravjev, P.: Binary factorization in Hopfield-like neural networks: single-step approximation and computer simulations. In: *Neural Networks World*, pp. 139–152 (2004)
2. Frolov, A.A., Húsek, D., Muravjev, P., Polyakov, P.: Boolean Factor Analysis by Attractor Neural Network. *Neural Networks, IEEE Transactions* 18(3), 698–707 (2007)
3. Földiák, P.: Forming sparse representations by local anti-Hebbian learning. *Biological cybernetics* 64(22), 165–170 (1990)
4. Fierro, R.D., Jiang, E.P.: Lanczos and the Riemannian SVD in information retrieval applications. *Numerical Algebra with Applications* 3(1), 1–18 (2003)
5. Berry, M., Dumais, S., Letsche, T.: Computational Methods for Intelligent Information Access. In: *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, San Diego, California, USA (1995)
6. Berry, M.W., Fierro, R.D.: Low-Rank Orthogonal Decomposition for Information Retrieval Applications. *Numerical Algebra with Applications* 1(1), 1–27 (1996)
7. Shahnaz, F., Berry, M., Pauca, P., Plemmons, R.: Document clustering using nonnegative matrix factorization. *Journal on Information Processing and Management* 42, 373–386 (2006)
8. Kolda, T.G., O’Leary, D.P.: Computation and uses of the semidiscrete matrix decomposition. *ACM Transactions on Information Processing* (2000)
9. Turk, M., Pentland, A.: Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 3(1), 71–86 (1991)
10. Spratling, M.W.: Learning Image Components for Object Recognition. *Journal of Machine Learning Research* 7, 793–815 (2006)
11. Frolov, A.A., Husek, D., Polyakov, P., Rezankova, H.: New Neural Network Based Approach Helps to Discover Hidden Russian Parliament Votting Paterns. In: *Proceedings of International Joint Conference on Neural Networks*, pp. 6518–6523. Omnipress, Madison, WI, USA (2006)