INDEPENDENT COMPONENT ANALYSIS FOR BINARY DATA: AN EXPERIMENTAL STUDY

Johan Himberg

Nokia Research Center
P.O. Box 407, 00045 NOKIA GROUP
Finland

Aapo Hyvärinen

Neural Networks Research Center
Helsinki University of Technology
P.O. 5400, 02015 HUT, Finland

ABSTRACT

We consider a mixing model where independent binary components are mixed using binary OR operations. Using extensive simulations, we investigate whether the model can be estimated using ordinary cumulant-based ICA algorithms. We show that the model can indeed be estimated if the data is sparse enough. We also compare the 3rd and 4th order cumulants. In the no-noise and low-noise cases, the 3rd order cumulant performs better, but in the presence of strong noise, the 4th-order cumulant, somewhat surprisingly, performs better for very sparse data.

1. INTRODUCTION

Independent component analysis (ICA) [3, 4] is a statistical model where the observed data is expressed as a linear transformation of latent variables that are nongaussian and mutually independent. In the classic version of the model, we have continuous-valued variables that are mixed linearly:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is the vector of observed random variables, $\mathbf{s} = (s_1, s_2, ..., s_n)^T$ is the vector of the independent latent variables (the "independent components"), and $\mathbf{A}$ is an unknown constant matrix, called the mixing matrix. The problem is then to estimate both the mixing matrix $\mathbf{A}$ and the realizations of the latent variables s_i, using observations of $\mathbf{x}$ alone. Exact conditions for the identifiability of the model were given in [1]; the most fundamental is that the independent components s_i must be nongaussian [1].

In many applications, the multivariate data $\mathbf{x}$ is binary or has strong binary nature, see e.g., [5, 6]. The linear mixing model in (1) can then no longer be used as is, because the linear mixing is not restricted to binary values. Below, we formulate an alternative model with purely binary operations. The validity of the application of ordinary ICA algorithms on such data is not obvious, either.

One approach would be to formulate new estimation methods and algorithms for a purely binary mixing model. However, since a lot of research has been conducted on ICA algorithms for continuous-valued data, it would be very useful if the ordinary algorithms could be used on binary data. In this paper, we investigate this possibility. This is done by simulations since a theoretical treatment seems too difficult. We use an ordinary cumulant-based ICA algorithm (FastICA) for binary data. We show that this works successfully if the data is *sparse* enough, i.e., most of the data values are zero. We also compare the performances of 3rd and 4th order cumulants (skewness and kurtosis).

2. BINARY DATA MODEL

Let $\mathbb{B}$ be the set of binary numbers $\{0, 1\}$. The mixing matrix $\mathbf{A}$ is an $m \times n$ matrix whose columns, the *basis vectors*, are binary vectors $\mathbf{a}_j \in \mathbb{B}^m$, $j = 1, 2, \ldots, n$. The independent source signal vectors are $\mathbf{s} \in \mathbb{B}^n$ where n is the number of sources, and the observed signal vectors are $\mathbf{x} \in \mathbb{B}^m$ where m is the number of signals. The basic linear ICA model 1 is replaced by the Boolean expression

$$x_i = \bigvee_{j=1}^{n} a_{ij} \wedge s_j, \quad i = 1, 2, \ldots, m \tag{2}$$

where $\wedge$ is Boolean AND and $\vee$ Boolean OR.

Instead of using Boolean operators Eq. 2 could be written using the linear mixing model and a non-linearity, for example,

$$\mathbf{x} = U(\mathbf{A}\mathbf{s}) \tag{3}$$

where $U(\mathbf{r})$ is a unit step function for vector $\mathbf{r}$ in $\mathbb{R}^d$ defined as

$$U(\mathbf{r}) = (u(r_1), u(r_2), \ldots, u(r_d))^T,$$

$$\text{where } u(r_i) = \begin{cases} 1 & \text{if } r_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

Finding the source signals will not be a trivial task even if the mixing matrix $\mathbf{A}$ is known since the step function U in Eq. 3 is not invertible. This situation is similar to the cases of noisy data or overcomplete bases. Only the more difficult task of finding the basis vectors is discussed in this paper; the estimation of the source signals could be performed by relatively simple maximum likelihood methods as with noisy data.

Intuitively, if $\mathbf{s}$ is sparse enough the observed signals should not be very different whether the data is generated by Eq. 1 or Eq. 3. This assumption justifies the following heuristics for estimating the binary matrix $\mathbf{A}$ using some algorithm for standard linear ICA: The mixing matrix is estimated assuming the linear ICA model. This gives an estimate $\hat{\mathbf{A}}_L$ of the mixing matrix $\mathbf{A}_L$ for the linear problem. To obtain an estimate of $\mathbf{A}$ that is binary, we use thresholding of the initial estimate $\hat{\mathbf{A}}_L$:

$$\hat{\mathbf{A}} = U(\mathbf{\Lambda}\hat{\mathbf{A}}_L - \mathbf{T}) \tag{4}$$

The diagonal scaling matrix $\mathbf{\Lambda}$ has elements

$$\lambda_i = \text{signmax}(\hat{\mathbf{a}}_i), \text{ where}$$

$$\text{signmax}(\mathbf{r}) = \begin{cases} \max(\mathbf{r}) & \text{if } |\max(\mathbf{r})| > |\min(\mathbf{r})| \\ |\min(\mathbf{r})| & \text{otherwise.} \end{cases}$$

where $\max(\mathbf{r})$ and $\min(\mathbf{r})$ mean taking maximum and minimum element of vector $\mathbf{r}$, respectively. The matrix $\mathbf{T}$ contains thresholds. Here we set its elements $t_{ij} = 0.5$ for all i, j.

3. EXPERIMENTS

Now, we perform extensive experiments to see if our heuristic method (estimating binary ICA with ordinary ICA algorithms) might work.

As mentioned above, the sparseness of the data may be very important for the success of the estimation. Thus, different amount of sparseness both in the mixing matrix and the source signals are used.

An important question is the choice of the objective function for ICA estimation. There are indications that using skewness instead of kurtosis might be a choice for doing ICA on this type of data [5]. However, the two functions have different qualitative behavior, so the situation may not be clear-cut: see Fig. 1. Here we use 3rd and 4th order cumulants with FastICA.

Further, we investigate the effect of noise on the estimation.

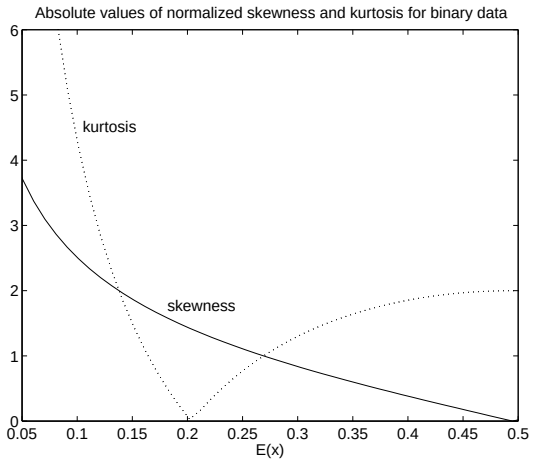


Figure 1: Normalized kurtosis and skewness for a binary variable as a function of the expectation of the variable. Their behavior is clearly different. This implies that for certain sparseness of binary data finding extremes of kurtosis might be more difficult than for skewness and vice versa.

3.1. Data generation

The mixing matrix $\mathbf{A}$ is generated randomly, but it is not allowed to have any zero basis vectors nor pair of identical basis vectors to prevent singularity. Therefore, candidate columns $\mathbf{a} = (a_1, a_2, \ldots, a_m)^T \in \mathbb{B}^m$ are generated one by one and checked against the existing ones: A candidate that is identical to an existing one, or zero, is rejected and a new candidate is generated. The elements in a candidate vector are generated independently from each other, given the probabilities of zero and one. The probability $p_a = P(a_i = 1)$ is same for all $i = 1, 2, \ldots, m$ for all columns $\mathbf{a}$ of the mixing matrix $\mathbf{A}$. The true expected ratio for an element of $\mathbf{A}$ being 1, $p_A = E(a..)$ averaged over all i, j, differs slightly from p_a due to the condition of nonsingularity.

The source signal vectors are binary random vectors $\mathbf{s} = (s_1, s_2, \ldots, s_n)^T \in \mathbb{B}^n$ whose variables are generated by independent binomial distributions that have probability $p_{sj} = P(s_j = 1)$ for $j = 1, 2, \ldots n$. A realization of source signals is generated by setting $p_{sj} = P(s_j = 1), j = 1, \ldots, n$ and generating an $n \times N$ binary matrix $\mathbf{S} = (\mathbf{s}(1), \mathbf{s}(2), \ldots, \mathbf{s}(N))$ For simplicity, we set the same probability $p_{sj} = p_s$ for all j in our tests.

Eq. 2 corresponds to the basic noise-free ICA model. However, we experiment also using noise-corrupted signals. The noise vectors are random binary vectors $\mathbf{e} = (e_1, e_2, \ldots, e_m)^T \in \mathbb{B}^m$ whose variables are generated by independent binomial distributions that have

the same probability $p_e = P(e_i = 1)$ for all signals $i = 1, 2, \ldots, m$. A realization of noise signals is created by generating a $m \times N$ binary matrix $\mathbf{E}$. The noise is added to the model output signals by bitwise exclusive-or operation $\oplus$. If a noise bit is 1 the corresponding signal bit will be flipped otherwise it remains unchanged.

A sample of noise corrupted signals $\mathbf{x}$ is generated by $\mathbf{X} = \mathbf{X_0} \oplus \mathbf{E}$ where $\mathbf{X_0} = U(\mathbf{AS})$ is the noise-free model output. The probability p_e is set so that a given noise level is achieved. The noise level is measured as the ratio between the number of "noise-on" bits in the noise signals $\mathbf{E}$ and "signal-on" bits in the noise-free signal $\mathbf{X_0}$:

$$NL = 100\% \frac{\sum_{t=1}^{N} \sum_{i=1}^{m} e(t)_i}{\sum_{t=1}^{N} \sum_{j=1}^{m} x_0(t)_i}. \quad (5)$$

3.2. Parameter selection

Experiments were run using 10 and 40 sources (n) for basic ($m = n$), underdetermined ($m < n$) and overdetermined ($m > n$) problems where m is the number of observed signals. The number of data samples was always set to $N = 100m$. For both cases 12 different combinations of prior source and mixing matrix densities p_s and p_a and three different noise levels (NL) were used, see Tab. 1. The data was randomly generated 30 times for each parameter combination. This meant altogether 6480 data sets.

Table 1: Parameter combinations for data generation

n	m	p_a	p_s	NL
10	8	0.2	0.05	0
	10	0.3	0.10	5
	15	0.4	0.20	25
			0.30	
40	30	0.2	0.01	0
	40	0.3	0.05	5
	60	0.4	0.10	25
			0.15	

3.3. ICA algorithm

The implementation that was used in the test was the FastICA package [2]. The symmetrical approach was used. For each data set, the algorithm was started from a random initialization and iterated for maximum 200 steps. This was repeated maximum five times if the algorithm did not converge within the step limit. The algorithm was applied to each data set using both kurtosis and skewness as contrast functions.

4. RESULTS

The performance was evaluated by counting the relative amount of correctly retrieved basis vectors. This is marked $R_{\%} = \frac{c}{n}100\%$ where c is the number of the estimated basis vectors $\hat{\mathbf{a}}_i$ of $\hat{\mathbf{A}}$ that are unique[1] and identical to some column $\mathbf{a}_i$ in the original $\mathbf{A}$. Note that an underdetermined problem has maximum $R_{\%} = 100\frac{m}{n}\%$, $m < n$. On the other hand, it is likely that an overdetermined problem gives higher scores than the basic problem since a large number of correct basis vectors may be found just by chance. If the algorithm did not converge on some test within the limits explained in previous section, $R_{\%} = 0$ was set for that run.

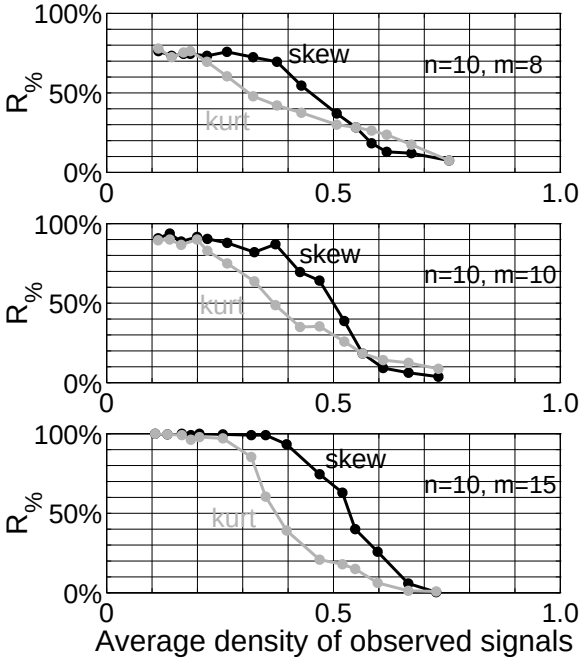
The results are presented in Figs. 2(a-d). These show the average performance as a function of the output signal density (sparseness), i.e., the average frequency of ones in the observed signals. The curves are computed by dividing the density values into 15 bins having the same number of samples. Since there are 12 combinations of source and mixing matrix densities and 30 trials for each combination, there are 24 samples for calculating one dot on the curve. More precisely, the dots are located at points (x_i, y_i) where x_i is the mean of the output densities of i-th bin and y_i is the mean of $R_{\%}$ in that bin, respectively.

In the noiseless case, we see that on average, skewness performs better than than kurtosis between output signal densities $0.3 \ldots 0.5$. Outside this region the contrast functions seem to give similar results. We found that $R_{\%}$ is only slightly lower for low noise level $NL = 5\%$, and the relative difference between skewness and kurtosis remains the same. Accordingly, the plot for $NL = 5\%$ has been left out for reasons of clarity. However, for high noise level $NL = 25\%$ the results are clearly different. Naturally, it appears that the share of correctly retrieved basis vectors is lower in general, but interestingly, kurtosis now seems to perform better, when the output signal is very sparse.

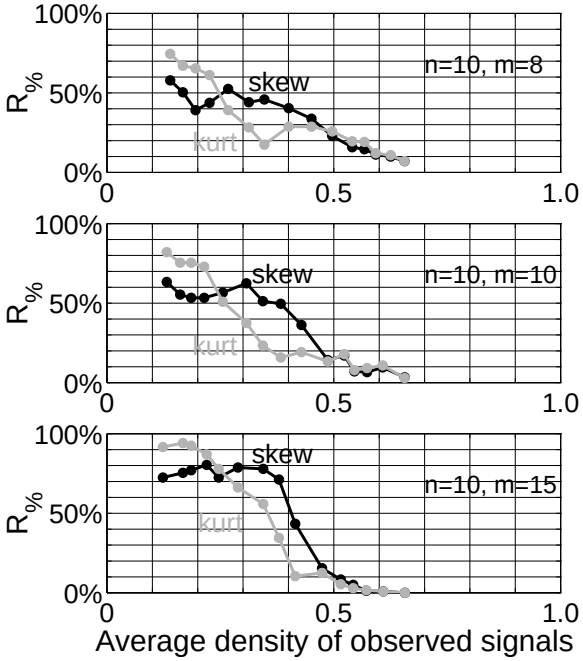
5. CONCLUSION

We investigated the feasibility of the application of ordinary ICA algorithms for purely binary data. Binary multivariate data come up in special applications, for example, in document retrieval [5]. Our experiments suggest that the basic linear ICA model can be used to form the binary basis vectors for the model described by Eq. 2 if the signals are sparse. The experiments sug-

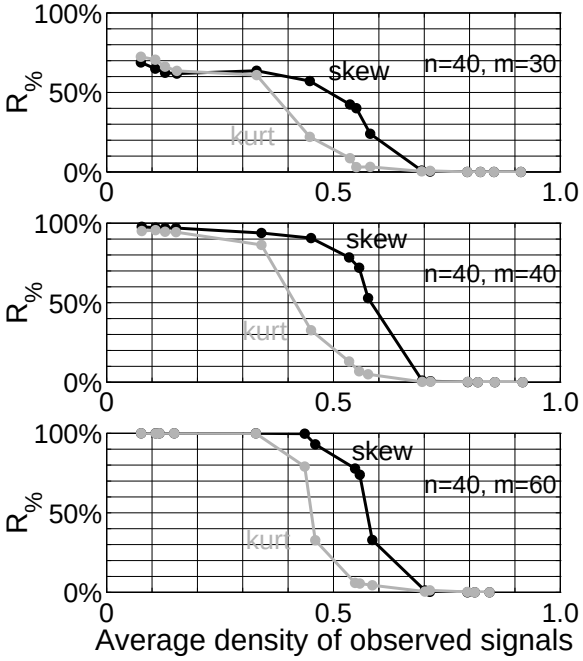
[1]The columns of the original mixing matrix are unique, but the same does not necessarily apply to the estimated binary mixing matrix.



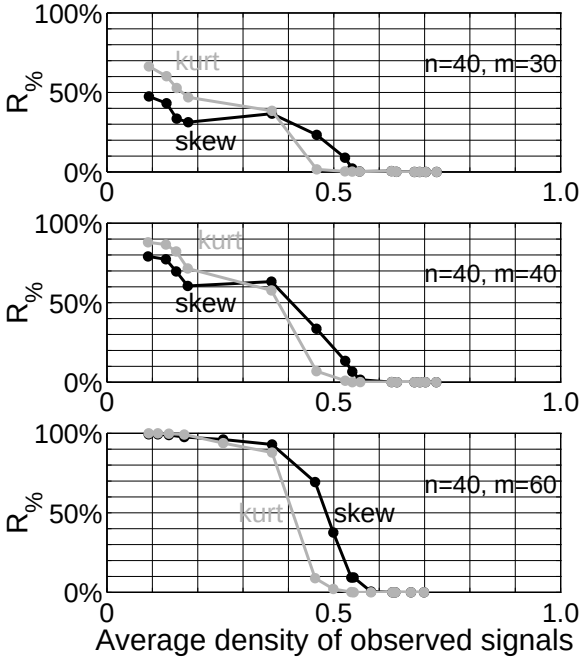
(a) NL=0%, $n = 10$



(b) NL=25%, $n = 40$



(c) NL=0%, $n = 10$



(d) NL=25%, $n = 40$

Figure 2: Relative amount of correctly retrieved basis vectors vs. average density of output signals. Gray curve show the average success percentage $R_{\%}$ for kurtosis as and black for skewness, respectively. Panels (a) and (b) refer to tests with 10 sources for two different noise levels (NL) and panels (c) and (d) for 40 sources, respectively. Each panel is divided into three subfigures where n shows the number of sources and m the number of observed signals.

gest also that skewness works better as contrast function for this kind of data on a certain range of model output density. A surprising result was that kurtosis seems to work better for very noisy and sparse data.

It is presumable that one could develop specialized algorithms that take advantage of the binary structure of the data and give better results in some cases. Our results suggest, however, that the use of well-known ICA algorithms can be extended to the purely binary ICA model when the data is sparse enough — without elaborating special algorithms.

6. REFERENCES

[1] P. Comon. Independent component analysis—a new concept? *Signal Processing*, 36:287–314, 1994.

[2] The FastICA MATLAB package. Available at `http://www.cis.hut.fi/projects/ica/fastica/`.

[3] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley Interscience, 2001.

[4] C. Jutten and J. Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.

[5] A. Kaban and M. Girolami. Clustering of text documents by skewness maximization. In *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, pages 435–440, Helsinki, Finland, 2000.

[6] J. Mäntyjärvi, J. Himberg, P. Korpipää, and H. Mannila. Extracting the Context of a Mobile Device User. In *Proc. of 8th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human-Machine System*, 2001. To appear.