# AN ITERATIVE NONLINEAR GAUSSIANIZATION ALGORITHM FOR RESAMPING DEPENDENT COMPONENTS

*Jen-Jen Lin**, *Naoki Saito, and Richard A. Levine*

Division of Statistics and Department of Mathematics
University of California, Davis, CA 95616, USA
emails: jjlin@mcu.edu.tw, saito@math.ucdavis.edu, levine@wald.ucdavis.edu

## ABSTRACT

We propose an Iterative Nonlinear Gaussianization Algorithm (INGA), which seeks a nonlinear map from a set of dependent random variables to independent Gaussian random variables. A direct motivation of the INGA is to extend the principal component analysis (PCA) which transforms a set of correlated random variables into uncorrelated (independent up to second order) random variables. An obvious advantage of deriving independent components is that we can simulate a stochastic process of dependent multivariate variables by sampling univariate independent variables. The quality of the transformation is evaluated by statistical tests on the Kullback-Leibler (KL) distance between the distribution of the transformed variables the standard multivariate Gaussian distribution $N(0, I)$. The quality of the simulations is evaluated *quantitatively* by the statistics of the KL distances between the sample mean distribution of the original samples and that of the simulated samples. Several numerical examples including synthetic and real-life image databases show the capabilities and limitations of INGA.

## 1. INTRODUCTION

Given $n$ samples of a dependent random vector $\mathbf{X}$ of $p$ dimensions ($n > p$), we are interested in resampling the dependent components in such a way that the resampled data and the observed samples obey the same $p$-variate distribution. In the case of one dimension (aside from the dependence), the classical bootstrap [1] "random resampling with replacement" has been a popular choice. Another direction is to use copulas [2], which join multivariate distributions to their one-dimensional marginal distributions. In the high dimensional case, however, both methods have limitations. Only linear models have been used in bootstrap analyses [1], and the copula-based algorithms are hard to apply for problems of dimension greater than 3 [2].

Independently from these resampling ideas, the method of Independent Component Analysis (ICA) [3] has become very popular, in particular, in the neural network community [4]. The objective of ICA is to find a linear transformation so that the given random vector can be represented "as independent as possible." Their motivations originally came from the problems of blind source separation and blind deconvolution (see e.g., [5] for a comprehensive review).

We propose a statistical algorithm called the Iterative Nonlinear Gaussianization Algorithm (INGA) — an extension to PCA. INGA tries to nonlinearly transform a set of correlated random variables to the standard multivariate Gaussian $N(0, I)$, at a similar computational cost to PCA, in an attempt to minimize the statistical dependence among the transformed coordinates. The difference between INGA and ICA lies in two aspects, although both seek statistically-independent coordinate systems. First, INGA seeks a *nonlinear* transform whereas ICA seeks a linear one. Second, the motivation of INGA is really *resampling* and *simulation* rather than blind source separation and blind deconvolution.

There are two parts to INGA: the forward and backward processes. The forward process iteratively transforms a given set of dependent random variables by: 1) applying PCA to decorrelate the random components; 2) matching their one-dimensional marginal distributions to the standard Gaussian distribution $N(0, 1)$; and 3) transforming the components linearly to improve the joint Gaussianity. At each iteration, the closeness of the resulting transformed variables to the standard joint Gaussian variates $N(0, I)$ is checked by statistical tests evaluated with the empirical P-value of certain distance measures such as the Kullback-Leibler distance, multivariate skewness, and kurtosis. The backward process generates new samples which presumably obey the same distribution as the original samples at our disposal. Section 2 and 3 discuss these in details.

Evaluation of the resamples or simulated data is often done subjectively by visually comparing them with the original samples. This subjective evaluation is particularly dominating in the area of texture modeling and simulation (e.g., [6], [7]). In order to *objectively* evaluate the similarity (or difference) between the original samples and the generated resamples, we use the Kullback-Leibler distance [8], which is summarized in Section 4.

Our motivation to develop INGA lies in image modeling and simulation from the given samples without knowledge of the underlying true probability model. However, the most difficult problem in image modeling is the "curse of dimensionality." In particular, a reliable estimate of probability density functions (pdf's) of high dimensional data

Current address: Department of Statistics, Ming Chuan University, Taipei, Taiwan.

This work was supported partially by NSF grants DMS-99-73032 and DMS-99-78321.

from a finite (and potentially small) number of samples are hard to obtain in general. Also, INGA itself is a relatively expensive algorithm (i.e., a constant multiple of the cost of PCA). Therefore, before actually applying INGA to image samples, we need to efficiently compress images (dimension reduction). Either PCA or wavelet transforms can be used for this dimension reduction. However, it is important to realize that PCA achieves only decorrelation and the wavelet transforms in general produce highly statistically dependent coefficients across different scales [6]. (See [9] and references therein for more about sparsity and statistical independence.)

Therefore, it doubly makes sense to apply INGA on the compressed representations of the original data. This compression strategy is always employed in this paper if the original dimension $p$ exceeds about 50. We present such an example in Section 5.

## 2. ITERATIVE NONLINEAR GAUSSIANIZATION ALGORITHM (INGA)

### 2.1. The INGA forward process

Let $\mathbf{X} = (X_1, \ldots, X_p)'$ be a random (column) vector of interest, which obeys a cumulative distribution function (cdf) $F_{\mathbf{x}}$. Let $\mathbf{x}$ be a realization (or an observed version) of this random vector. Let $\Phi$ denote the cdf of the standard $p$-variate Gaussian distribution $N(0, I_p)$, where $I_p$ is the $p \times p$ identity matrix. The INGA forward process consists of the following steps:

(i) `Initialize` $\mathbf{x}^{(0)} = \mathbf{x}$ and set $k = 0$.

(ii) `Apply` PCA to the images, i.e., $\mathbf{y} = B'\mathbf{x}^{(k)}$, where $B$ is an orthogonal matrix for decorrelating $\mathbf{x}^{(k)}$.

(iii) `Transform` $\mathbf{u} = F_{\mathbf{y}}(\mathbf{y})$ so that each of the $p$ random variates $\mathbf{u} = (u_1, \ldots, u_p)'$ is uniformly distributed on the interval $(0, 1)$.

(iv) `Transform` $\mathbf{z} = \Phi^{-1}(\mathbf{u})$ so that each of the $p$ random variates $\mathbf{z} = (z_1, \ldots, z_p)'$ obeys the standard Gaussian distribution $N(0, 1)$.

(v) `Transform` $\tilde{\mathbf{z}} = A\mathbf{z}$ through a linear operator $A$, such that $\tilde{\mathbf{z}}$ obeys the standard multivariate Gaussian distribution $N(0, I_p)$.

(vi) `Check` convergence to the multivariate Gaussian distribution. If satisfactory convergence not attained, increment $k$, set $\mathbf{x}^{(k)} = \tilde{\mathbf{z}}$, and go to (ii).

Step (iv) does not guarantee us to produce *standard multivariate* Gaussian random variates $N(0, I_p)$ although each marginal distribution is the standard Gaussian $N(0, 1)$. This is why we need to apply the transformation of Step (v) to make the random variates more like standard multivariate Gaussian.

The operator $A$ in Step (v) is obtained by minimizing the $L^2$ error between the characteristic function of $\tilde{\mathbf{z}}$ and that of the standard multivariate Gaussian $N(0, I_p)$. Section 3 describes how to compute such an operator $A$ in detail.

The variables transformed by $A$ may be neither truly multivariate Gaussian nor independent at a given iteration.

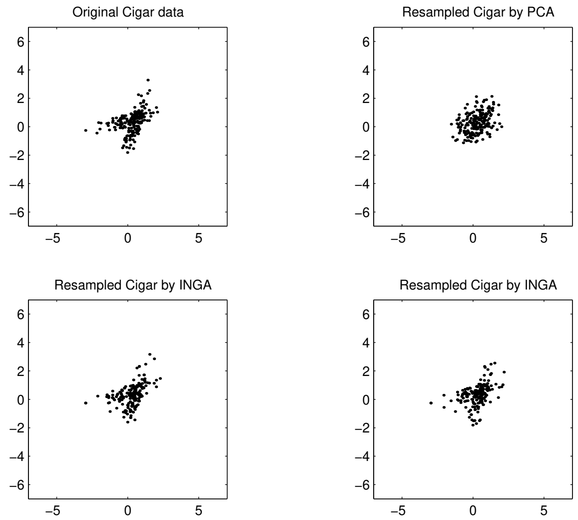


Figure 1: Resampling of the "cigar" data by PCA and INGA. Counter-clockwise from the top left: Original samples; Resamples from PCA coordinates; Resamples by INGA; Yet another set of resamples by INGA.

We thus need to iterate over the $\tilde{\mathbf{z}}$ variates until the maximum correlation of the components of $\tilde{\mathbf{z}}$ (i.e., the size of the off-diagonal components of the covariance matrix of $\tilde{\mathbf{z}}$) becomes less than a given tolerance $\epsilon$, and the Monte Carlo estimates of the multivariate skewness, kurtosis, and negentropy test of multivariate Gaussianity [8] reach satisfactory levels. Section 4 describes these validation procedures in detail.

### 2.2. INGA backward process

With the records of the INGA forward process (i.e., storing all the information about $A$, $F_{\mathbf{y}}$ for each iteration), we can go backward: starting from the INGA coefficients, we can get the original sample $\mathbf{x}$ by reversing the forward steps (ii)–(v). If we assume that the final INGA coefficients obey the standard multivariate Gaussian, it is easy to generate a new set of such coefficients. We then can go backwards to synthesize the data in the starting coordinates (i.e., resampling).

(i) `Generate` INGA coefficients.

(ii) `Invert` all the iterations of the INGA forward process to synthesize the data in the starting coordinates.

Although Section 5 will present several applications of the INGA, it is motivating to show a simple example at this point. Figure 1 displays data generated from the infamous "cigar" data, two obliquely overlapping bivariate Gaussian distributions. Resampling of this data under the PCA coordinates performs poorly and fails to capture the two cigar shapes. On the other hand, the INGA successfully simulates this distribution (See Section 5 for the quantitative analysis of this simulation).

## 3. HOW TO COMBINE THE MARGINALLY GAUSSIAN DATA TO FORM JOINTLY GAUSSIAN DATA?

### 3.1. Theoretical analysis of Step (iv)

After we have the marginal Gaussian variables $Z_i$ in Step (iii) of the INGA forward process, Step (iv) finds a linear transform $A$ so that $\tilde{\mathbf{Z}} = A\mathbf{Z}$ closely obeys the standard multivariate Gaussian distribution. How can we find such an $A$? Let $\phi(t_i) = e^{-\frac{1}{2}t_i^2}$ be the characteristic function of $Z_i$ and let $f(A|\mathbf{t}) = E(e^{i\mathbf{t}'\tilde{\mathbf{Z}}}) = E(e^{i\mathbf{t}'A\mathbf{Z}})$ be the characteristic function of $\tilde{\mathbf{Z}}$. Then, we define a cost function, which is an $L^2$ error between these two characteristic functions:

$$D(A) = \left\| f(A|\mathbf{t}) - \prod_{i=1}^{p} \phi(t_i) \right\|^2 . \tag{1}$$

The operator $A$ is obtained by minimizing $D(A)$. A straightforward way to minimize (1) is to use the perturbation of $D(A)$ about $A_0$, the orthogonal matrix used in PCA to decorrelate the $Z_i$'s:

$$A = A_0 + \delta A \tag{2}$$

where $\delta A$ is a perturbation matrix with components $\delta a_{ij}$ which are the elements obtained from the Taylor expansion of $D(A)$ about $A_0$,

$$\delta\mathbf{a} = -H^{-1}(A_0) \cdot \frac{\partial D}{\partial \mathbf{a}}(A_0), \tag{3}$$

where $\frac{\partial D}{\partial \mathbf{a}} = (\frac{\partial D}{\partial a_{11}}, \frac{\partial D}{\partial a_{12}}, \cdots, \frac{\partial D}{\partial a_{pp}})'$, and $H$ is the $p^2 \times p^2$ Hessian matrix,

$$H = \begin{pmatrix} \frac{\partial^2 D}{\partial a_{11}\partial a_{11}} & \frac{\partial^2 D}{\partial a_{11}\partial a_{12}} & \cdots & \frac{\partial^2 D}{\partial a_{11}\partial a_{pp}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial^2 D}{\partial a_{pp}\partial a_{11}} & \frac{\partial^2 D}{\partial a_{pp}\partial a_{12}} & \cdots & \frac{\partial^2 D}{\partial a_{pp}\partial a_{pp}} \end{pmatrix} .$$

Although this perturbation is straightforward, one can easily see that this becomes quickly impractical even for moderate sizes of $p$.

### 3.2. Numerical minimization of $D(A)$

There is an additional challenge for numerically minimizing (1) via the perturbation in (2) and (3). Let us first rewrite $D(A)$ in (1) as

$$D(A) = \int \left| f(A|\mathbf{t}) - \prod_{i=1}^{p} \phi(t_i) \right|^2 dt_1 \cdots dt_p. \tag{4}$$

To find the minimizer $A$ to (4), we need numerical evaluation of both (4) and (3). To perform these, we use the Gaussian quadrature method, i.e., a weighted sum of the integrand evaluated at the nodes $t_{i,j}, i = 1, \ldots, p, j = 1, \ldots, J$ of $\mathbf{t}$, where $J$ is the required number of nodes. This sum can be written as

$$D(A) \approx \sum_{j_1,\cdots,j_p} w_{j_1,\cdots,j_p} \left| f(A|t_{1,j_1}, \ldots, t_{p,j_p}) - \prod_{i=1}^{p} \phi(t_{i,j_i}) \right|^2 . \tag{5}$$

This approximation is also used to compute the quantities in (3). In this problem, we use the Gauss-Chebyshev quadrature method, which simply uses the nodes $-.538$; 0; .538 in each dimension in (5). For the two-dimensional case, the required number of nodes is nine. For the general $p$ dimensional case, $\mathbf{t} = (t_1, \cdots, t_p)$, we need $3^p$ nodes. This numerical approach is again only feasible for small to moderate $p$. One reason is that the limitation of INTEGER numbers in the computer is $(2^{31}) - 1 = 2.1475e + 009$ which is even smaller than $3^{20} = 3.4868e + 009$ ($p = 20$). The other reason is the CPU time: in the case of $p = 20$ dimensions, the estimated CPU time is 2.71 days; for $p = 100$, it is out of question.

Instead of finding $A$ by going through the $3^p$ nodes all at once, which we call the regular optimization process, we use a *sequential* process to find $A$ *approximately*. This sequential process consists of the following steps:

**[Initial Step]** Let $\mathbf{Z}_2 = (Z_1, Z_2)'$ be the first two components of $\mathbf{Z}$. Find the $2 \times 2$ matrix $A_2 = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ by minimizing the cost function (5) using the regular optimization process. Then the transformed vector $\tilde{\mathbf{Z}}_2 = A_2 \mathbf{Z}_2$ has a two-dimensional distribution with mean zero and identity covariance matrix.

**[General Step,** $m \geq 3$] Once $A_{m-1}$ and $\tilde{\mathbf{Z}}_{m-1}$ are obtained, append $Z_m$ to $\tilde{\mathbf{Z}}_{m-1}$ to form $\mathbf{Z}_m = (\tilde{\mathbf{Z}}'_{m-1}, Z_m)' = (\tilde{Z}_1, \cdots, \tilde{Z}_{m-1}, Z_m)'$. Then, look for the linear transformation with the special structure $\tilde{A}_m = \begin{pmatrix} I_{m-1} & 0 \\ \mathbf{a}'_{m-1} & a_{mm} \end{pmatrix}$ where $\mathbf{a}'_{m-1} = (a_{m1}, \cdots, a_{m,m-1})$. The transformed random vector $\tilde{\mathbf{Z}}_m = \tilde{A}_m \mathbf{Z}_m$ then minimizes

$$D(\tilde{A}_m) = \left\| f(\tilde{A}_m|\mathbf{t}_m) - \prod_{i=1}^{m} \phi(t_i) \right\|^2 , \tag{6}$$

for which we make the approximation

$$f(\tilde{A}_m|\mathbf{t}_m) = E(e^{i\mathbf{t}'_m\tilde{\mathbf{Z}}_m}) \approx E(e^{i\mathbf{t}'_{m-1}\tilde{\mathbf{Z}}_{m-1}})E(e^{it_m\tilde{Z}_m}) ,$$

where $\mathbf{t}_m = (t_1, \cdots, t_m)'$. This is the *key* assumption, but is reasonable because $\tilde{\mathbf{Z}}_m$ is eventually expected to be the standard multivariate Gaussian. The last row vector $\mathbf{a}'_m = (\mathbf{a}'_{m-1}, a_{mm})$ of $\tilde{A}_m$ in (6) can be found by minimizing

$$\tilde{D}(\mathbf{a}_m) = \| E(e^{it_m(\mathbf{a}'_{m-1}\tilde{\mathbf{Z}}_{m-1} + a_{mm}Z_m)}) - e^{-\frac{1}{2}t_m^2} \|^2 . \tag{7}$$

Since only one characteristic variable $t_m$ is involved, we need only to find $\mathbf{a}_m$ by one-dimensional 3-point Gaussian quadrature using the perturbation $\mathbf{a}_m = \mathbf{a_0}_m + \delta\mathbf{a}_m$ in a similar manner as (2), where $\mathbf{a_0}_m = (0, 0, \cdots, 0, 1)'$. Combining this optimal $\mathbf{a}_m$ with the previous $A_{m-1}$, we have the $m \times m$ optimal transformation matrix with the structure

$$A_m = \begin{pmatrix} A_{m-1} & 0 \\ \mathbf{a}'_{m-1} & a_{mm} \end{pmatrix} . \tag{8}$$

Note the relationship $\tilde{\mathbf{Z}}_m = \tilde{A}_m \mathbf{Z}_m = A_m(Z_1, \cdots, Z_m)'$. Section 5 demonstrates the validity of the sequential optimization process by comparing it with the regular optimization process using simple examples. This step is repeated until $m = p$. For more details, see [10].

## 4. MODEL VALIDATION MEASURES AND STRATEGIES

In essence, INGA builds a probability model of the underlying distribution for the data of interest. This model is used to synthesize/simulate new samples, which can be used for a variety of important tasks, such as diagnostics and classification. It is then of critical importance to know the accuracy of our model in a quantitative manner.

The Kullback-Leibler (KL) distance [11], also known as the relative entropy or cross-entropy, is a measure of the difference between two probability models, and is defined as $J(p_{\mathbf{X}}, q_{\mathbf{X}}) = \int p_{\mathbf{X}}(\mathbf{u}) \log \frac{p_{\mathbf{X}}(\mathbf{u})}{q_{\mathbf{X}}(\mathbf{u})} \, d\mathbf{u}$, where $p_{\mathbf{X}}$ and $q_{\mathbf{X}}$ are two pdf's of interest. The KL distance between the model distribution and the standard Gaussian distribution is called the neg-entropy. We use the KL distance in two ways in INGA and its applications. One is to check the closeness of the transformed variables to the standard multivariate Gaussian variables in Step (vi) of the INGA forward process by estimating the neg-entropy. The other is to quantify the closeness of the obtained distribution and the original unknown distribution via the generated resamples and the original observed samples.

The difficulty in these validations lies in the estimation of the KL distance. Jones and Sibson [12] and Comon [3] independently showed the neg-entropy for an univariate random variable can be approximated by a function of their cumulants using an Edgeworth expansion. The neg-entropy thus can be estimated from the sample estimates of the corresponding cumulants. For higher dimensional cases, we derived the sample estimates of neg-entropy as well as those of the general KL distances between the two pdf's using an Edgeworth expansion in [8].

To evaluate the quality of the resamples using the KL distance, there is still one problem: The Edgeworth-based estimation of the KL distance only applies to distributions not far from Gaussian [8]. Hence we estimate the KL distance of the the sample mean distributions rather than the distributions of the original samples and resamples.

## 5. NUMERICAL EXAMPLES

We now demonstrate the capabilities and limitations of INGA using several examples ranging from simple synthetic stochastic processes to a real-life image database. All of the simulations using INGA work as follows:

[**Step 1 (optional)**] Compress the original data using PCA (or wavelets or any other sparsifying transformation).

[**Step 2**] Apply INGA to generate new samples relative to the coordinates used in Step 1.

[**Step 3 (optional)**] Rotate the new samples back to the original (i.e., standard) coordinate system.

Step 1 and 3 are necessary if the dimensionality of the original data is high (e.g., $p > 50$). These steps are necessary in the eye database example below.

### 5.1. Three-dimensional cigar data for justification of the sequential optimization process

In order to investigate the stability of the sequential optimization process discussed in Section 3, we constructed the three-dimensional cigar data (i.e., three anisotropic Gaussian distributions overlapping in $\mathbf{R}^3$) as follows: 1) Create samples of the random variable $\mathbf{Y} = (Y_1, Y_2, Y_3)'$ with $Y_1 \sim N(0,1)$, $Y_2 \sim \text{Unif}(0,1)$, and $Y_3 \sim \text{Gamma}(3)$; 2) Apply three different rotations $R_{\pi/12,0}, R_{5\pi/12,0}, R_{0,\pi/12}$ to $\mathbf{Y}$ ($R_{\theta,\phi}$ represents a rotation with angle $\theta$ in the plane specified by the first two coordinates and angle $\phi$ in the plane specified by the last two coordinates in $\mathbf{R}^3$); and 3) Collapse three components of each rotated version of $R_{\theta_i,\phi_i}\mathbf{Y}$ to a new coordinate $X_i$, $i = 1,2,3$ to create a new random vector $\mathbf{X}$. Now the samples of $\mathbf{X}$ are overlapping obliquely. We chose $p = 3$ since we really wanted to compare the performance between the sequential and regular optimization processes. For this $p = 3$ case, the regular optimization process is computationally feasible since it only requires $27 \, (= 3^3)$ Gauss-Chebyshev quadrature nodes. As we discussed in Section 4, the neg-entropy was used to measure the distance between the transformed variables and the standard multivariate Gaussian. Figure 2 shows the neg-entropy of the regular and sequential optimization processes during the first 30 iterations. The neg-entropy difference between the sequential and regular processes are essentially gone after 20 iterations.

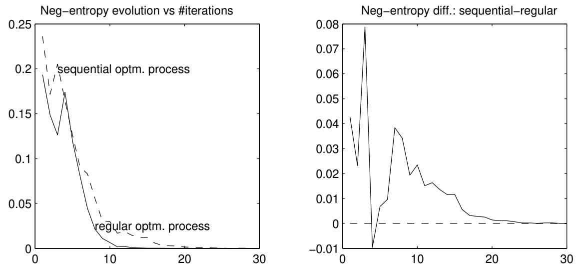


Figure 2: Neg-entropy of the regular and sequential optimization processes during the first 30 INGA iterations. Right figure shows the difference between these two neg-entropy values.

### 5.2. Examples for testing the independence

To perform the test of the standard multivariate Gaussianity (i.e., this also implies the test of independence), we use the Monte Carlo method to obtain the empirical P-value of multivariate skewness and kurtosis, and neg-entropy. Figure 3 shows these empirical P-values for the cigar data ($p = 2$) and eye data ($p = 5, 15, 25$) (see Subsection 5.4 for the details of the eye data). The multivariate kurtosis and skewness for $N(0, I_p)$ are 3 and 0, respectively. From these plots, we may conclude that the INGA forward process successfully transforms the original random variables to independent standard Gaussian variables within four to five iterations.

### 5.3. The spike process

The spike process—although it is a very simple stochastic process—can be used to check the performance of the various resampling/simulation techniques in a very clear man-

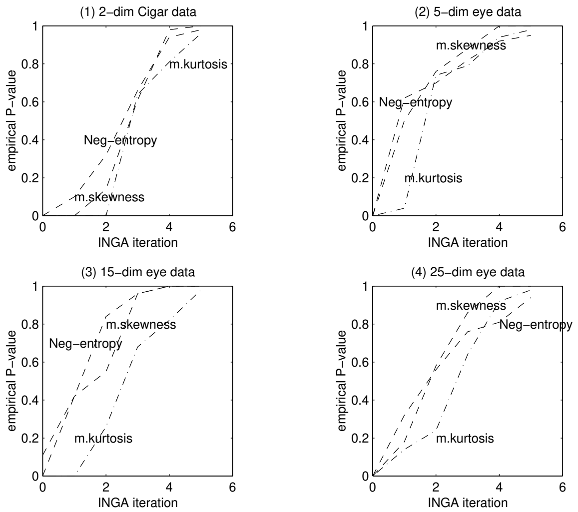

Figure 3: Empirical P-values of multivariate kurtosis, skewness, and neg-entropy of the INGA simulations of two-dimensional cigar data and eye data with various compression rates ($p = 5, 15, 25$).

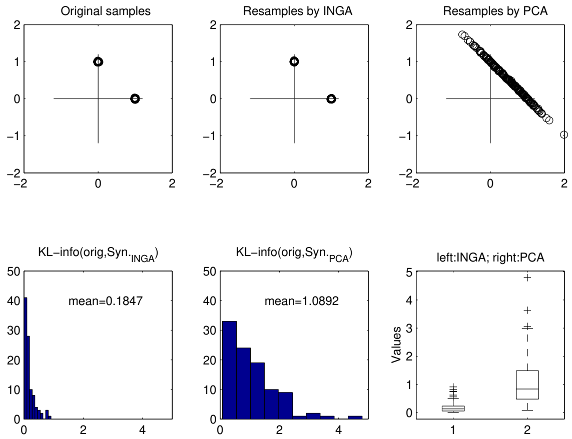

Figure 4: The spike process simulation ($p = 2$). The first row from left to right: original samples; resamples by INGA; resamples by PCA. The second row: a histogram of KL distances between the original samples and the INGA resamples; the corresponding histogram for the PCA resamples; their boxplots.

ner. A $p$-dimensional version of this process is described as follows: $\mathbf{X}(\omega) = \mathbf{e}_\omega$ where $\omega$ is uniformly and randomly chosen from the set of integers $\{1, \dots, p\}$, and $\mathbf{e}_k$ is simply the $k$th standard basis vector of $\mathbf{R}^p$. So, a unit spike is randomly shifting among the locations $k = 1, \dots, p$. It is known that for $p = 2$ we can obtain the independent coordinates by a simple linear transformation (in fact, a 45 degree rotation), but for $p > 2$ no linear transformation can provide the independent coordinates [9]. Therefore, it is very interesting to see how INGA performs for this process with various $p$.

Figure 4 shows the results for $p = 2$. In this case, INGA obtained the independent coordinates perfectly after only one iteration whereas resampling on the PCA coordinates failed. We note that a resampling strategy using the PCA is to: 1) Transform the original samples into the PCA coordinates; 2) Resample each coordinate by computing its empirical cdf; and 3) Rotate the resamples back to the original coordinates. This works if the original data truly obey the multidimensional (correlated) Gaussian distribution; only in that case is the second step of resampling each coordinate justified.

Figure 4 also shows histograms and boxplots of the KL distances between the original samples and 100 resamples by INGA and PCA using the sample mean distributions. The the mean KL distance between the original and the INGA is 0.1847 whereas that of the PCA is 1.0892, respectively. (If the original samples and the resamples are distributed in exactly the same manner, then the KL distance would be 0.)

For the spike process of $p = 32$, the results are shown in Figure 5. In this case, after two iterations of INGA, things did not really improve. One can observe that: 1) INGA could not obtain the truly independent coordinates; 2) simulation based on PCA is worse than that by INGA. Yet, the KL distances between the original samples and the INGA resamples (its mean value = 15.3965) are much closer than those between the originals and the PCA resamples (the mean value = 92.10).

### 5.4. Eye image database

The original eye database contains 100 images of size $12 \times 12$ pixels. Figure 6 compares the simulations by INGA and PCA. In both cases, the images are first reduced to 25 dimensions by PCA. The simulations by INGA and PCA visually look similar. However, we can clearly see the quantitative difference once we compute the KL distances between the original samples and these simulations. The mean KL distance between the originals and the INGA simulations is 3.1781 whereas that between the originals and the PCA simulations is 19.6943.

## 6. CONCLUSION

We have presented a new nonlinear algorithm to seek statistically independent coordinates—INGA. Although INGA does not guarantee to produce truly independent coordinates, we have demonstrated that the INGA-based simulations of the dependent variables are quantitatively superior to the PCA-based simulations through several examples ranging from synthetic stochastic processes to a real-life eye image database. We also demonstrated the importance of the accessibility to the distributions of the KL distances between the original samples and the simulated samples via the sample mean distribution since this is one of the very few ways to measure the quality of the simulations in a quantitative manner. We are currently comparing the performance of INGA with that of ICA.

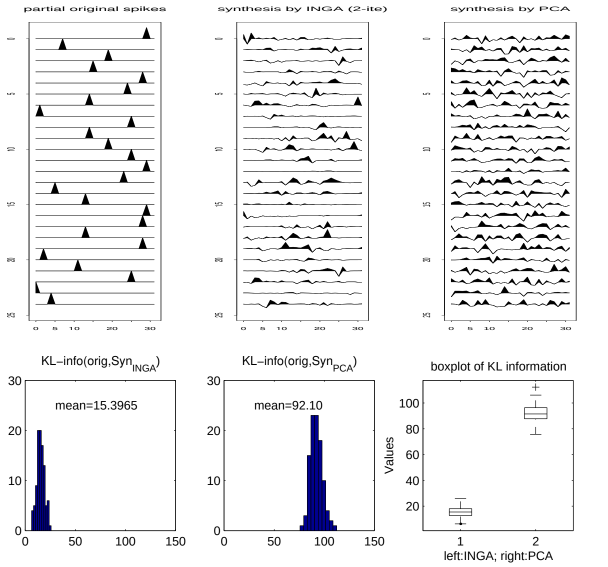


Figure 5: The spike process simulation ($p = 32$). The first row from left to right: original samples; resamples by INGA; resamples by PCA. The second row: a histogram of KL distances between the original samples and the INGA resamples; the corresponding histogram for the PCA resamples; their boxplots.

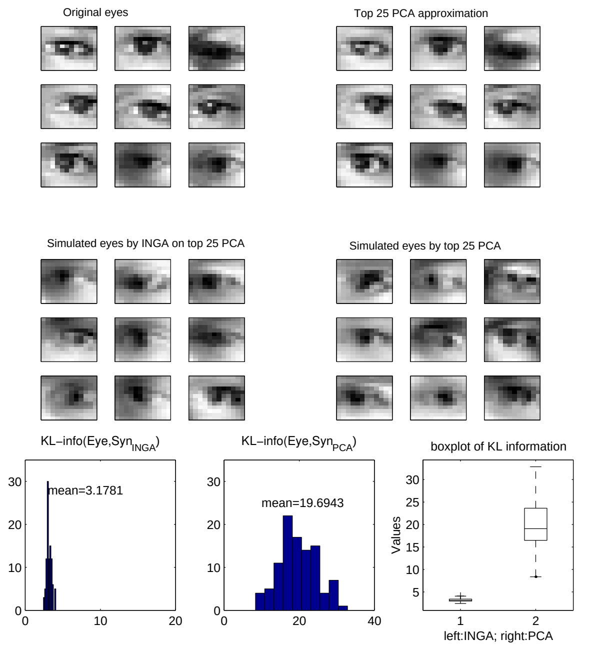


Figure 6: Simulations of the eye image database by INGA and PCA. The first row from left to right: original image samples; their top 25 PCA approximations. The second row: simulated eyes by INGA; simulated eyes by PCA. The third row: a histogram of KL distances between the originals and INGA simulations; the corresponding histogram for INGA; their boxplots.

## 7. REFERENCES

[1] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. New York: Chapman & Hall, Inc., 1993.

[2] R. B. Nelsen, *An Introduction to Copulas*, vol. 139 of *Lecture Notes in Statistics*. Springer-Verlag, 1998.

[3] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[4] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[5] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proc. IEEE*, vol. 90, pp. 2009–2025, 1998.

[6] J. Portilla and E. P. Simoncelli, "Texture modeling and synthesis using joint statistics of complex wavelet coefficients." IEEE Workshop on Statistical and Computational Theories of Vision, Fort Collins, CO, June 22nd 1999.

[7] S. C. Zhu, Y. Wu, and D. Mumford, "FRAME: Filters, Random fields And Maximum Entropy–Toward a unified theory for texture modeling," *Intern. J. Comput. Vision*, vol. 27, no. 2, pp. 1–20, 1998.

[8] J.-J. Lin, N. Saito, and R. A. Levine, "Edgeworth expansions of the Kullback-Leibler information," tech. rep., Division of Statistics, Univ. California, Davis, 1999. submitted to J. Amer. Statist. Assoc.

[9] N. Saito and B. Larson, "Sparsity vs statistical independence from a best-basis view point." In preparation.

[10] J.-J. Lin, N. Saito, and R. A. Levine, "An iterative nonlinear Gaussianization algorithm and its applications." In preparation.

[11] S. Kullback, *Information Theory and Statistics*. New York: John Wiley & Sons, 1959. Republished by Dover in 1997.

[12] M. C. Jones and R. Sibson, "What is projection pursuit? (with discussion)," *J. R. Statist. Soc. A*, vol. 150, no. Part 1, pp. 1–36, 1987.