Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 962105, 13 pages
doi:10.1155/2012/962105

*Research Article*

# Measuring Non-Gaussianity by Phi-Transformed and Fuzzy Histograms

**Claudia Plant,[1] Son Mai Thai,[2] Junming Shao,[3] Fabian J. Theis,[4] Anke Meyer-Baese,[1] and Christian Böhm[2]**

[1] *400 Dirac Science Library, Florida State University, Tallahassee, FL 32306-4120, USA*
[2] *Department for Informatics, Research Unit for Database Systems, University of Munich, Oettingenstraße 67, 80538 Munich, Germany*
[3] *Klinikum rechts der Isar der TUM, Ismaninger Straße 22, 81675 Munich, Germany*
[4] *Helmholtz Zentrum München, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany*

Correspondence should be addressed to Claudia Plant, cplant@fsu.edu

Received 14 February 2012; Accepted 1 April 2012

Academic Editor: Juan Manuel Gorriz Saez



Independent component analysis (ICA) is an essential building block for data analysis in many applications. Selecting the truly meaningful components from the result of an ICA algorithm, or comparing the results of different algorithms, however, is nontrivial problems. We introduce a very general technique for evaluating ICA results rooted in information-theoretic model selection. The basic idea is to exploit the natural link between non-Gaussianity and data compression: the better the data transformation represented by one or several ICs improves the effectiveness of data compression, the higher is the relevance of the ICs. We propose two different methods which allow an efficient data compression of non-Gaussian signals: Phi-transformed histograms and fuzzy histograms. In an extensive experimental evaluation, we demonstrate that our novel information-theoretic measures robustly select non-Gaussian components from data in a fully automatic way, that is, without requiring any restrictive assumptions or thresholds.

## 1. Introduction

Independent component analysis (ICA) is a powerful technique for signal demixing and data analysis in numerous applications. For example, in neuroscience, ICA is essential for the analysis of functional magnetic resonance imaging (fMRI) data and electroencephalograms (EEGs). The function of the human brain is very complex and can be only imaged at a very coarse spatial resolution. Millions of nerve cells are contained in a single voxel of fMRI data. The neural activity is indirectly measured by the so-called BOLD-effect, that is, by the increased supply of active regions with oxygenated blood. In EEG, the brain function can be directly measured by the voltage fluctuations resulting from ionic current flows within the neurons. The spacial resolution of EEG, however, is even much lower than that of fMRI. Usually, an EEG is recorded using an array of 64 electrodes distributed over the scalp. Often, the purpose of acquiring fMRI or EEG data is obtaing a better understanding of brain function while the subject is performing some task. An example for such an experiment is to show subjects images while they are in the scanner to study the processing of visual stimuli, see Section 4.1.4. Recent results in neuroscience, for example [1], confirm the organization of the human brain into distinct functional modules. During task processing, some functional modules are actively contributing to the task. However, many other modules are also active but not involved into task-specific activities. Due to the low resolution of fMRI and EEG data, we observe a partial volume effect: the signal at one particular voxel or electrode consists of task-related activities, nontask-related activities, and a lot of noise. ICA is a powerful tool for signal demixing and, therefore, in principle very suitable to reconstruct the interesting task-related activity.

Many ICA algorithms use the non-Gaussianity as implicit or explicit optimization goal. The rationale behind this decision is due to a reversion of the central limit theorem: the sum of a sufficiently large number of independent random

variables, each with finite mean and variance, will approximate a normal distribution. Therefore, an algorithm for the demixing of signals has to optimize for non-Gaussianity in order to obtain the original signals. We adopt this idea in this paper and define a data compression method which yields a high compression rate exactly if the data distribution is far away from Gaussianity and no compression in the data distribution is exactly gaussian.

However, the evaluation and interpretation of the result of ICA is often difficult for two major reasons. First, most ICA algorithms always yield a result, even if the underlying assumption (e.g., non-Gaussianity for the algorithm FastICA [2]) is unfulfilled. Thus, many ICA algorithms extract as many independent sources as there are mixed signals in the dataset, no matter how many of them really fulfill the underlying assumption. Second, ICA has no unique and natural evaluation criterion to assess the relevance or strength of the detected result (like, e.g., the variance criterion for principle component analysis (PCA)). Different ICA algorithms use different objective functions, and to select one of them as an *overall objective*, or *neutral* criterion would give unjustified preference to the result of that specific algorithm. Moreover, if the user is interested in comparing an ICA result to completely different modeling techniques like PCA, regression, mixture models, and so forth, these ICA-internal criteria are obviously unsuitable. Depending on the actual intension of the user, different model selection criteria for ICA might be appropriate. In this paper, we investigate the *compressibility* of the data as a more neutral criterion for the quality of single component or the overall ICA result.

## 2. Related Work

*2.1. Model Selection for ICA.* Model selection for ICA or automatically identifying the most interesting components is an active research question. Perhaps the most widely used options for model selection are measures like Kurtosis, Skewness, and approximations of neg-entropy [3]. However, these measures are also applied as optimization criteria by some ICA algorithms. Thus, a comparison of the results across algorithms is impossible. Moreover, these measures are very sensitive with respect to noise points and single outliers.

In [4], Rasmussen et al. propose an approach for model selection of epoched EEG signals. In their model order selection procedure, the data set is split into two sets, training- and test set, to ensure an unbiased measure of generalization. With each model hypothesis, the negative logarithm of the likelihood function is then calculated using a probabilistic framework on the training and test set. The model having minimal generalization error is selected. This approach, however, is based on certain assumptions about source autocorrelation and tends to be sensitive to noise.

The most common method for model order selection is based on principal component analysis (PCA) of the data covariance matrix, which is proposed by Hyvärinen et al. [3]. The choice of number of sources to be selected is based on the number of dominant eigenvalues which significantly contribute to the total variance. This approach is fast and simple to implement, however, it suffers from a number of problems, for example, an inaccurate eigenvalue decomposition of the data covariance matrix in the noise-free case with fewer numbers of sources than sensors and sensitivity to noise. Moreover, there are no reasons to say that the subspace spanned by dominant principal components contains the source of interest [5]. Another approach proposed by James and Hesse [5] is to do the step-wise extraction of the sources until it reaches a predefined accuracy. However, the choice of reasonable accuracy level is also one drawback of this algorithm.

Related to model selection but still a different problem is the reliability of ICA results. The widely used iterative fix-point algorithm FastIca [6] converges towards different local optima of the optimization surface. The technique Icasso [7] combines Bootstrapping with a visualization to allow the user to investigate the relationship between different ICA results. Reliable results can be easily identified as dense clusters in the visualization. However, no information on the quality of the results is provided, which is the major focus of our work. Similar to Icasso, Meinecke et al. [8] proposed a resampling method to assess the quality ICA results by computing the stability of the independent subspaces. First, they create surrogate datasets by randomly selecting independent components from an ICA decomposition and apply the ICA algorithm for each of the surrogate data sets. Then, they separate the data space into one or multidimensional subspaces by their block structure and compute the uncertainty for each subspace. This proposed reliability estimation can be used to choose the appropriate BSS-model, to enhance the separation performance and, most importantly, to flag components which have a physical meaning.

*2.2. Minimum Description Length for Model Selection.* The minimum description length (MDL) principle is based on the simple idea that the best model to describe the data is one with the overall shortest description of the data and model itself, and it is essentially the same as Occam's razor.

The MDL principle has been successfully applied for model selection for a large variety of tasks, ranging from linear regression [9], image segmentation [10] to polyhedral surface models [11].

In data mining, the MDL principle has recently attracted some attention enabling parameter-free algorithms to graph mining [12], clustering, for example [13–15], and outlier detection [16]. Sun et al. [12] proposed GraphScope, a parameter-free technique to mine information from streams of graphs. This technique used MDL to decide how and when to form and modify communities automatically. Böhm et al. [15] proposed OCI, a novel fully automatic algorithm to clustering non-Gaussian data with outliers, based on MDL to control the splitting, filtering, and merging phase in a parameter-free and very efficient top-down clustering approach. CoCo [16], a technique for parameter-free outlier detection, is based on the ideas of data compression and coding costs. CoCo used MDL to define an intuitive outlier factor together with a novel algorithm for outlier detection.

This technique is parameter free and can be applied to a wide range of data distributions. OCI combines local ICA with clustering and outlier filtering. Related to this idea, in [17], Gruber et al. propose an approach for automated image denoising combining local PCA or ICA with model selection by MDL.

In this paper, we propose a model selection criterion based on the MDL principle suitable for measuring the quality of single components as well as complete ICA results.

## 3. Independent Component Analysis and Data Compression

One of the fundamental assumptions of many important ICA algorithms is that independent sources can be found by searching for maximal non-Gaussian directions in the data space. Non-Gaussianity leads to a decrease in entropy, and therefore, to a potential improvement of the efficiency of data compression. In principle, the achievable compression rate of a dataset, after ICA, is higher compared to the original dataset. The principle of minimum description length (MDL) uses the probability $P(x)$ of a data object $x$ to represent it according to Huffman coding. Huffman coding gives a lower bound for the compression rate of a data set $D$ achievable by any concrete coding scheme, as follows: $\sum_{x\in D} -\log_2(P(x))$. If $x$ is taken from a continuous domain (e.g., the vector space $\mathcal{R}^d$ for the blind source separation of a number $d$ of signals), the relative probability given by a probability density function $p(x)$ is applied instead of the absolute probability. The relative and absolute log-likelihoods (which could be obtained by discretizing $x$) are identical up to a constant value which can be safely ignored, as we discuss in detail in Section 3.1. For a complete description of the dataset (allowing decompression), the parameters of the probability density function (PDF) such as mean and variance for Gaussian PDFs need to be coded and their code lengths added to the negative log-likelihood of the data. We call this term the code book. For each parameter, a number of bits equal to $(1/2)\log_2(n)$ where $n$ is the number of objects in $D$, is required, as fundamental results from information theory have proven [18]. Intuitively, the term $(1/2)\log_2(n)$ reflects the fact that the parameters need to be coded more precisely when a higher number $n$ of data objects is modeled by the PDF. The MDL principle is often applied for model selection of parametric models like Gaussians, or Gaussian mixture models (GMMs). Gaussian mixture models vary in the model complexity, that is, the number of parameters needed for modeling. MDL-based techniques are well able to compare models of different complexity. The main purpose of the code book is to punish complexity in order to avoid overly complex, over-fitted models (like a GMM having one component exactly at the position of each data object: such a model would yield a minimal Huffman coding, but also a maximal code-book length). By the two concepts, Huffman coding using the negative log-likelihood of a PDF and the code-book for the parameters of the PDF, the principle of MDL provides a very general framework which allows the comparison of very different modeling techniques like principal component analysis (based on a Gaussian PDF model), clustering [13], regression [19] for continuous domains, but, in principle, also for discrete or mixed domains. At the same time, model complexity is punished and, therefore, overfitting avoided. Related criteria for general model selection include, for example, the Bayesian information criterion and the Aikake information criterion. However, these criteria are not adapted to the ICA model. In the following section, we discuss how to apply the MDL principle in the context of ICA.

*3.1. General Idea of the Minimum Description Length Principle.* The minimum description length (MDL) principle is a well-established technique for selecting the best model out of a finite or infinite number of possible models for a given data set $D$ (in our case, a signal). The model is usually given in terms of a probability function $f(x)$ which assigns to every element $x \in D$ a probability that this element occurs in the dataset. For continuous domains, $f(x)$ is a probability density function satisfying $\int_{-\infty}^{+\infty} f(x)\mathbf{d}x = 1$. The idea of MDL is that $f(x)$ can be used as a basis to compress the data set $D$ using Huffman coding and to exploit that this coding becomes the more efficient (w.r.t. the achievable compression rate, or more precisely the code length after compression) the better $f(x)$ represents the true data distribution. According to Huffman coding, the minimum code length corresponds to the negative log-likelihood of the data set, that is,

$$\mathrm{NLLH}_f(D) = -\sum_{x\in D} \log_2 f(x). \tag{1}$$

While only for discrete domains the values of $f(x)$ are scaled between 0 and 1, this negative log-likelihood is also applied for continuous domains, but then some caveats apply. Basically, we can always reduce the continuous case to the noncontinuous case by discretizing the data (e.g., by a regular grid with a fixed resolution $g$). In this case,

$$F_g(x) = \int_{g\cdot\lfloor x/g\rfloor}^{g\cdot\lfloor x/g+1\rfloor} f(\xi)\mathbf{d}\xi \tag{2}$$

is a probability function scaled between 0 and 1 with

$$\lim_{g\to 0} \frac{F_g(x)}{g} = f(x). \tag{3}$$

For the negative log-likelihood of the so-discretized dataset $D_g$, we get

$$\begin{aligned} \mathrm{NLLH}_f\left(D_g\right) &= -\sum_{x\in D} \log_2 F_g(x) \\ &\approx -\sum_{x\in D} \log_2 g \cdot f(x) \\ &= \mathrm{NLLH}_f(D) - n\log_2 g, \end{aligned} \tag{4}$$

where in the case $g \to 0$ we have exact equality and also $-n\log_2 g \to \infty$, corresponding to the obvious fact that we need an infinite number of bits to represent a real number with infinite precision. However, when comparing different

models of the data (i.e., different probability functions $f_1(x)$ and $f_2(x)$), we simply have to ensure that all data are basically discretized with the same (and sufficiently high) resolution $g$ in the original data space and then ignore the term $-n\log_2 g$ which is equal in all compared models of the data (note that data in a computer is always represented with finite precision, and, therefore, always implicitly discretized):

$$\begin{aligned}\lim_{g\to 0}&\left(\mathrm{NLLH}_{f_2}\left(D_g\right)-\mathrm{NLLH}_{f_1}\left(D_g\right)\right)\\ &=\lim_{g\to 0}\Big(\left(\mathrm{NLLH}_{f_2}(D)-n\log_2 g\right)\\ &\qquad-\left(\mathrm{NLLH}_{f_1}(D)-n\log_2 g\right)\Big)\\ &=\mathrm{NLLH}_{f_2}(D)-\mathrm{NLLH}_{f_1}(D).\end{aligned}\tag{5}$$

We simply have to observe that (1) the resolution is not implicitly changed by any transformations of the dataset (like, e.g., a linear scaling $\alpha\cdot x$ of the data objects) and (2) that ignoring the term might lead to negative values of $\mathrm{NLLH}_f(D)$ (so we partly lose the nice intuition of a number of bits encoding the data objects, and particularly that 0 is a lower limit of this amount of information).

The principle of coding a signal with a pdf $f(x)$ is visualized in Figure 1 where a signal (a superposition of three sinuses) is coded. To represent one given point of the signal (at $t = 15.5$), we should actually use a discretization of $x$ as indicated in the right part of the diagram to obtain an actual code length for the value $x$. However, we can also directly use the negative log-likelihood of the value $x$ with the above-mentioned implications.

In addition to the amount of information which is caused by the negative log-likelihood of the data, we need also to code the function $f$. From an information-coding perspective, we need this information in order to be able to decode the data again after transferring it through a communication channel. The function $f$ tells us which code words translate back to what original data objects, so it serves as a code book. From a statistical perspective, the coding of $f$ is needed to avoid overfitting. The intention of $f$ is to generalize the data, and not to anticipate it, as a weird function would do which has simply a peak at every position where a data object is available. For both purposes, the representation of the code book must require considerably less information than the data itself. In this paper, we will propose two different methods to represent the function $f$. In Figure 1, a kernel density estimator (KDE) was used. However, KDE needs a number of parameters which is the same as the number of points, and, therefore, it is not suitable for our purpose. The classical (parametric) method is to use a class of model functions (like a Gaussian pdf) and to code the parameters (for Gaussian, $\mu$ and $\sigma$) with an amount of information corresponding to $(1/2)\log_2 n$ per parameter. This number can be derived by an optimization process which takes into account that a small error in the parameters does not lead to a serious deterioration of the NLLH, particularly if $n$ is small and only a few data objects are modeled by $f$. The number of $(1/2)\log_2 n$ bits represents an optimal trade-off (and includes this deterioration of NLLH already). Throughout this paper, we will use the minimum description length of a dataset as goal for minimization:

$$\mathrm{MDL}_f(D)=\mathrm{NLLH}_f(D)+\frac{1}{2}\#\mathrm{PAR}(f)\cdot\log_2 n.\tag{6}$$

We will in the following sections propose nonparametric methods which are both related to histograms. Therefore, the number of parameters in principle corresponds to the number of bins. We do not use histograms directly since our goal is to code the signals in a way that punishes the Gaussianity and rewards the non-Gaussianity. Thus, we propose two different methods which modify the histogram concept in a suitable way.

*3.2. Phi-Transformed Histograms.* Techniques like [15] or [20] successfully use the exponential power distribution (EPD), a generalized distribution function including Gaussian, Laplacian, uniform, and many other distribution functions for assessing the ICA result using MDL. The reduced entropy of non-Gaussian projections in the data allows a higher compression rate and thus favors a good ICA result. However, the selection of EPD is overly restrictive. For instance, multimodal and asymmetric distributions cannot be well represented by EPD but are highly relevant to ICA. In the following, we describe an alternative representation of the PDF which is efficient if (and only if) the data is considerably different from Gaussian. Besides the non-Gaussianity, we have no additional assumption (like for instance EPD) on the data. To achieve this, we tentatively assume Gaussianity in each signal of length $n$ and transform the assumed Gaussian distribution into a uniform distribution in the interval $(0, 1)$ by applying the Gaussian cumulative distribution function $\Phi((x-\mu)/\sigma)$ (the $\Phi$-transformation) to each signal of the data representation to be tested (e.g., after projection on the independent components). Then, the resulting distribution is represented by a histogram $(H_1,\ldots,H_b)$ with a number $b$ of equidistant bins where $b$ is optimized as we will show later. $H_j(1\le j\le b)$ is the number of objects falling in the corresponding half open interval $[(j-1)/b,\ j/b)$. If the signal is Gaussian indeed, then the signals after $\Phi$-transformation will be uniform and the histogram bins will be (more or less) uniformly filled. Therefore, a trivial histogram with only one bin will in this case yield the best coding cost (and thus, no real data compression comes into effect). The $\Phi$-transformation itself causes a change of the coding cost which is equivalent to the entropy of the Gaussian distribution function, as we show in as follows.

The negative log-likelihood of the signal before the $\Phi$-transformation with the tentative assumption that the signals are compressed by the Gaussian pdf correspond to

$$\begin{aligned}\mathrm{NLLH}_{\mathrm{before}}&=\sum_{x\in D}-\log_2\left(\frac{1}{\sqrt{2\pi\sigma^2}}\mathbf{e}^{-(x-\mu)^2/2\sigma^2}\right)\\ &=n\cdot\log_2\left(\sqrt{2\pi\sigma^2}\right)+\frac{1}{2\sigma^2\ln 2}\sum_{x\in D}(x-\mu)^2,\end{aligned}\tag{7}$$

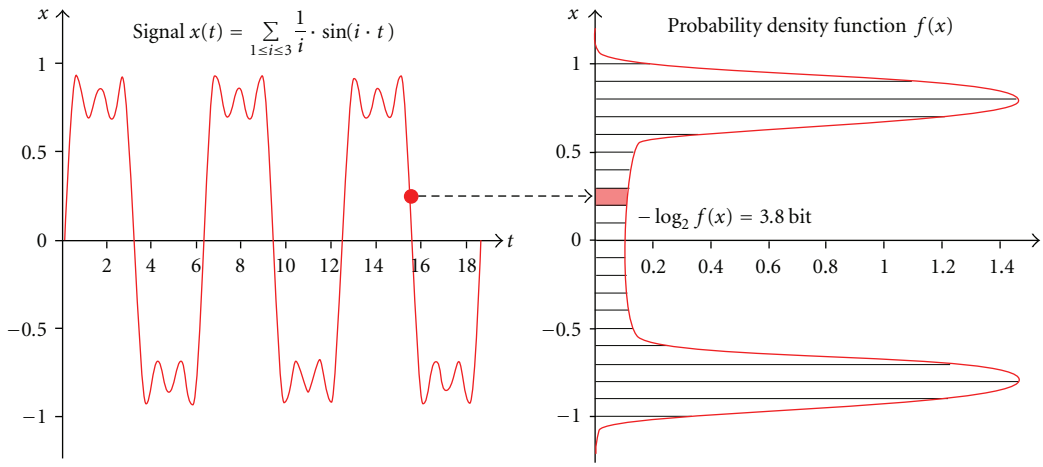



Figure 1: MDL-based compression of signals by Huffman coding.

and since $(1/n)\sum (x-\mu)^2$ is exactly the definition of the variance $\sigma^2$, we have

$$\mathrm{NLLH}_{\mathrm{before}} = n \cdot \log_2\left(\sqrt{2\pi \mathbf{e}\sigma^2}\right), \tag{8}$$

which is independent from the distribution which the signal $x$ actually has. After the $\Phi$-transformation, we code the signal under the assumption that it is uniformly distributed in $(0, 1)$. Thus, we obtain

$$\mathrm{NLLH}_{\mathrm{after}} = \sum_{x \in D} -\log_2(1) = 0, \tag{9}$$

and again, we do not worry that it appears as if no information is necessary to code the signals after the $\Phi$-transformation. But the difference between coding of the signals before and after the $\Phi$-transformation corresponds to

$$\mathrm{PRE} = \mathrm{NLLH}_{\mathrm{before}} - \mathrm{NLLH}_{\mathrm{after}} = n \cdot \log_2\left(\sqrt{2\pi \mathbf{e}\sigma^2}\right). \tag{10}$$

Representing the histogram $(H_1, \ldots, H_b)$ as a probability density function (integrating to 1) leads to

$$f_H(x) = H_{\lfloor b \cdot x + 1 \rfloor} \cdot \frac{b}{n}. \tag{11}$$

The negative log-likelihood of this $\Phi$-transformed signal corresponds to

$$\begin{aligned} \mathrm{NLLH}_{f_H}(D) &= \sum_{x \in D} -\log_2\left(H_{\lfloor b \cdot x + 1 \rfloor} \cdot \frac{b}{n}\right) \\ &= \sum_{x \in D} \log_2 \frac{n}{H_{\lfloor b \cdot x + 1 \rfloor}} - \sum_{x \in D} \log_2 b, \end{aligned} \tag{12}$$

and since we have $H_i$ objects in histogram bin $i$, we can change the first sum into the entropy of the histogram:

$$\mathrm{NLLH}_{f_H}(D) = \sum_{1 \le i \le b} \left(H_i \cdot \log_2 \frac{n}{H_i}\right) - n \cdot \log_2 b. \tag{13}$$

Using this coding scheme, the overall code length ($\mathrm{CLRG}(D, b)$, code length relative to Gaussianity) of the signal is provided by

$$\begin{aligned} \mathrm{CLRG}(D, b) = &\overbrace{\sum_{1 \le j \le b} H_j \cdot \log_2 \frac{n}{H_j}}^{\text{histogram entropy}} \\ &- \overbrace{n \cdot \log_2 b}^{\text{offset cost}} \\ &+ \overbrace{\frac{b-1}{2} \log_2 n}^{\text{code book}} \\ &+ \overbrace{\mathrm{PRE}}^{\text{preproc}}. \end{aligned} \tag{14}$$

As introduced in Section 3, the first two terms represent the negative log-likelihood of the data given the histogram. The first corresponds to the entropy of the histogram, and the second term, stemming from casting the histogram into a PDF, has also the following intuition: when coding the same data with a varying number of histogram bins, the resulting log-likelihoods are based on different basic resolutions of the data space (a grid with a number $b$ of partitions). Although the choice of a particular basic resolution is irrelevant for the end-result, for comparability, all alternative solutions must be based on a common resolution. We choose $g = 1$ as basic resolution, and subtract for each object the number of bits by which we know the position of the object more precisely than in the basic resolution. The trivial histogram having $b = 1$ represents the case where the data is assumed to be Gaussian: since the Gaussian cumulative distribution function $\Phi((x-\mu)/\sigma)$ has been applied to the data, Gaussian data are transformed into uniform data, and our histograms have an implicit assumption of uniformity *inside* each bin. Therefore, we call it *offset cost* because it stands for coding the position of a value *inside* a histogram bin. If some choice of $b \neq 1$ leads to smaller $\mathrm{CLRG}(D, b)$, we have evidence that the signal is different from Gaussian. It is easy to see that in

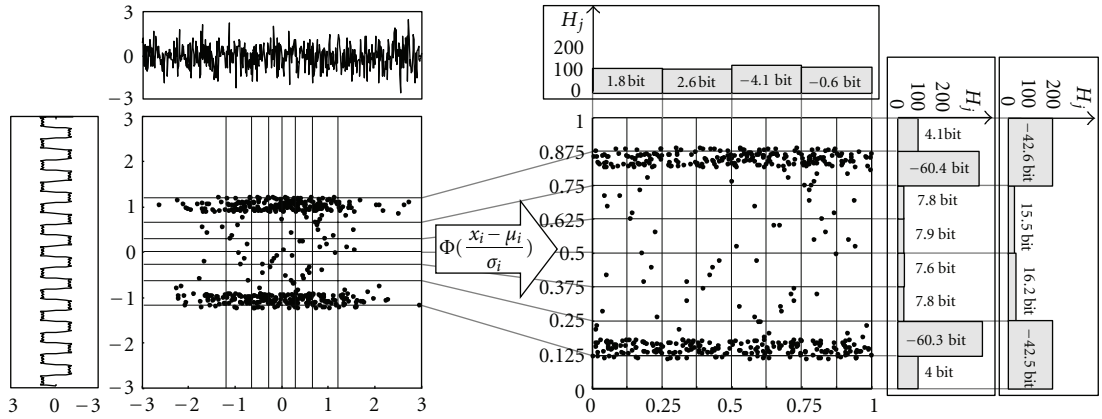



FIGURE 2: Overview of the computation of CLRG$(D, b)$: left: scatter plot of two signals in original space. On the $x$-axis: Gaussian noise signal. On the $y$-axis: signal with a rectangular pattern. The lines represent the quantiles assuming a Gaussian distribution; right: the scatter plot after applying the Gaussian CDF. The quantiles now form an equidistant grid. On the axes: histograms and compression costs using Huffman coding. Since the Gaussian signal does not contain any pattern beyond Gaussianity, it cannot be compressed in CDF-space.

the case $b = 1$ the code length is CLRG$(D, b)$ = PRE, which is a consequence of our definition of the offset cost. Therefore, we call our cost function code length relative to Gaussianity, (CLRG). If no choice that $b \neq 1$ leads to CLRG$(D, b) <$ PRE, then either the data is truly Gaussian or the number of data objects is not high enough to give evidence for non-Gaussianity. In the latter case, we use Gaussianity as the safe default-assumption. The third term is the cost required for the code book: to completely describe $b$ histogram bins it is sufficient to use $b - 1$ codewords since the remaining probability is implicitly specified. The last term, PRE is for preprocessing, that is, taking the $\Phi$-transform into account.

Figure 2 gives an overview and example of our method. On the left side, the result of an ICA run is depicted which has successfully separated a number $d = 2$ of signals (each having $n = 500$ points). The corresponding scatter-plot shows a Gaussian signal on the $x$-axis, a rectangular signal on the $y$-axis (note that the corresponding signal plots on the axes are actually transposed for better visibility). On the right side, we see the result after applying the Gaussian CDF. Some histograms with different resolutions are also shown. On the $x$-axis, the histogram with $b = 4$ bins is approximately uniformly filled (like also most other histograms with a different selection of $b$). Consequently, only a very small number of bits is saved compared to Gaussianity (e.g., only 4.1 bits for the complete signal part falling in the third bin $H_3$) by applying this histogram as PDF in Huffman coding (here, the cost per bin are reported including log-likelihood and offset-cost). The overall saving of 0.29 bit are contrasted by a code-book length of $(3/2)\log_2 n = 13.4$, so the histogram representation does not pay off. In contrast, the two histograms on the $y$-axis do pay off, since for $b = 8$, we have overall savings over Gaussianity of 81.3 bit by Huffman coding, but only $(7/2)\log_2 n = 31.4$ bits of codebook.

*3.2.1. An Optimization Heuristic for the Histogram Resolution.* We need to optimize $b$ individually for each signal such that the overall coding cost CLRG$(D, b)$ is minimized. As an efficient and effective heuristic, we propose to only consider histogram resolutions where $b$ is a power of 2. This is time efficient since the number of alternative results is logarithmic in $n$ (as we will show), and the next coarser histogram can be intelligently gained from the previous. In addition, the strategy is effective since a sufficient number of alternative results is examined.

We start with a histogram resolution based on the worst-case assumption that (almost) all objects fall into the same histogram bin of a histogram of very high resolution $b_m$. That means that the log-likelihood approaches 0. The offset cost corresponds to $-n\log_2 b_m$ but the parameter cost are very high: $((b_m - 1)/2)\log_2 n$. The other extreme case is the model with the lowest possible resolution $b = 1$ having no log-likelihood, no offset-cost, and no parameter cost. The histogram with resolution $b_m$ can pay off only if the following condition holds:

$$n\log_2 b_m \geq \frac{b_m - 1}{2}\log_2 n, \tag{15}$$

which is certainly true if $b_m \leq n/2$. We use $b_m = 2^{\lfloor \log_2 n \rfloor}$, the first power of two less or equal $n$ as starting resolution. Then, in each step, the algorithm generates a new histogram $H' = (H'_1, \ldots, H'_{b/2})$ from the previous histogram $H = (H_1, \ldots, H_b)$ by merging each pair of adjacent bins using $H'_j = H_{2j-1} + H_{2j}$ for all $j$ having $1 \leq j \leq b/2$. The overall number of adding operations for histogram bins starting from the histogram $H^{\text{start}} = (H_1^{\text{start}}, \ldots, H_{b_m}^{\text{start}})$ to the final histogram $H^{\text{end}} = (H_1^{\text{end}})$ corresponds to

$$\sum_{1 \leq i \leq b_m/2} i = b_m - 1 = 2^{\lfloor \log_2 n \rfloor} - 1 \in O(n). \tag{16}$$

The coding cost of the data with respect to each alternative histogram is evaluated as described in Section 3.2 and the histogram with resolution $b_{\text{opt}}$ providing the best compression is reported as result for dimension $i$. In the case of $b_{\text{opt}} = 1$, no compression was achieved by assuming

non-Gaussianity. After having optimized $b_{\text{opt}}$ for each signal separately, $\text{CLRG}(D, b)$, the coding costs of the data are provided as in Section 3.2 applying $b_{\text{opt}}$. To measure the overall improvement in compression achieved by ICA $\text{CLRG}(D, b)$ is summed up across all dimensions $i$:

$$\text{CLRG}(D) = \sum_{1 \le i \le d} \left( \min_{0 \le \log_2 b \le \lfloor \log_2 n \rfloor} \text{CLRG}(D, b) \right). \tag{17}$$

*3.3. Fuzzy Histograms.* Often histograms are not a good description of data since they define a discontinuous function whereas the original data distribution often corresponds to a continuous function. Since we want to focus on non-Gaussianity without any other assumption on the underlying distribution function, a good alternative to histograms is fuzzy histograms. In statistics, often kernel density estimators (KDEs) are applied in cases where a continuous representation of the distribution function is needed. However, KDEs require a number of parameters which is higher than the number of objects, and, therefore, KDEs are not suitable for our philosophy of compressing the dataset according to the defined distribution function (although other information-theoretic KDEs exist). Therefore, we apply the simpler fuzzy histograms which extend histograms as follows.

We have a kernel function $\kappa_{\mu_i,\sigma}(x)$ which is assigned to each fuzzy histogram bin $i$. Like with ordinary histograms, the location parameters $\mu_i$ are equidistant, that is,

$$\mu_i = m \cdot i + t, \tag{18}$$

and the scale parameter $\sigma$ is uniform for all bins (and also called bandwidth). In this paper, we use the normal distribution:

$$\kappa_{\mu_i,\sigma}(x) = n_{\mu_i,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left( -\frac{(x-\mu_i)^2}{\sigma^2} \right) \tag{19}$$

since it allows an elegant way to express the coding cost relatively to Gaussianity without explicitly transforming the dataset using the cumulative standard normal distribution $\Phi(x)$. To every histogram bin, a weight $w_i$, which indicates to which extent the bin is filled, is assigned. The sum over all $w_i$, is unity. The fuzzy histogram then defines the probability density function:

$$f(x) = \sum_{1 \le i \le b} w_i \cdot \kappa_{\mu_i,\sigma}(x), \tag{20}$$

which is continuous (as it is a sum of continuous functions) and integrates to 1 (as all $w_i$ sum up to one and each kernel function integrates to 1).

We determine the positions $\mu_1, \ldots, \mu_b$ (or, actually the parameters $m$ and $t$ to determine all $\mu_i$ in an equidistant way) as well as the bandwidth $\sigma$ in an iterative learning algorithm. We initialize the parameters such that

$$\begin{aligned} \mu_1 &= \min_{x \in D}(x), \qquad \mu_b = \max_{x \in D}(x), \\ w_i &= \frac{1}{b}, \quad (\forall i, 1 \le i \le b), \qquad \sigma = \frac{\mu_b - \mu_1}{b}. \end{aligned} \tag{21}$$

That means that we set the initial slope $m = (\mu_b - \mu_1)/b$ and $t = \mu_1 - m$.

Each point $x \in D$ may be assigned to more than one bin. It is gradually assigned and the sum of all assignments equals 1. The assignment is based on Bayes' theorem:

$$p(i \mid x) = \frac{w_i \cdot \kappa_{\mu_i,\sigma}(x)}{\sum_{1 \le j \le b} w_j \cdot \kappa_{\mu_j,\sigma}(x)}, \tag{22}$$

and the weights can be determined as

$$w_i = \frac{1}{|D|} \sum_{x \in D} p(i \mid x). \tag{23}$$

Then, we assign the points according to (22). We then determine each $\mu_i$ (calling it $\hat{\mu}_i$) individually (temporarily omitting the requirement that they are equi-distant) as

$$\hat{\mu}_i = \frac{1}{|D| \cdot w_i} \sum_{x \in D} p(i \mid x) \cdot x \tag{24}$$

and determine $m$ and $t$ as a weighted linear regression of the $\hat{\mu}_i$. Let $\overline{\mu} = \sum_{1 \le i \le b} w_i \cdot \hat{\mu}_i$ be the weighted average of all $\hat{\mu}_i$ and $\overline{i} = \sum_{1 \le i \le b} w_i \cdot i$ the weighted average of all $i$. Then, we obtain

$$m = \frac{\sum_{1 \le i \le b} w_i \cdot \left(i - \overline{i}\right) \cdot \left(\hat{\mu}_i - \overline{\mu}\right)}{\sum_{1 \le i \le b} w_i \cdot \left(i - \overline{i}\right)^2}, \qquad t = \overline{\mu} - m\overline{i}. \tag{25}$$

Finally, we determine the bandwidth parameter $\sigma$ by the average variance which is caused by $D$ in every bin:

$$\sigma^2 = \frac{1}{|D|} \sum_{x \in D} \sum_{1 \le i \le b} p(i \mid x) \cdot (x - \mu_i)^2. \tag{26}$$

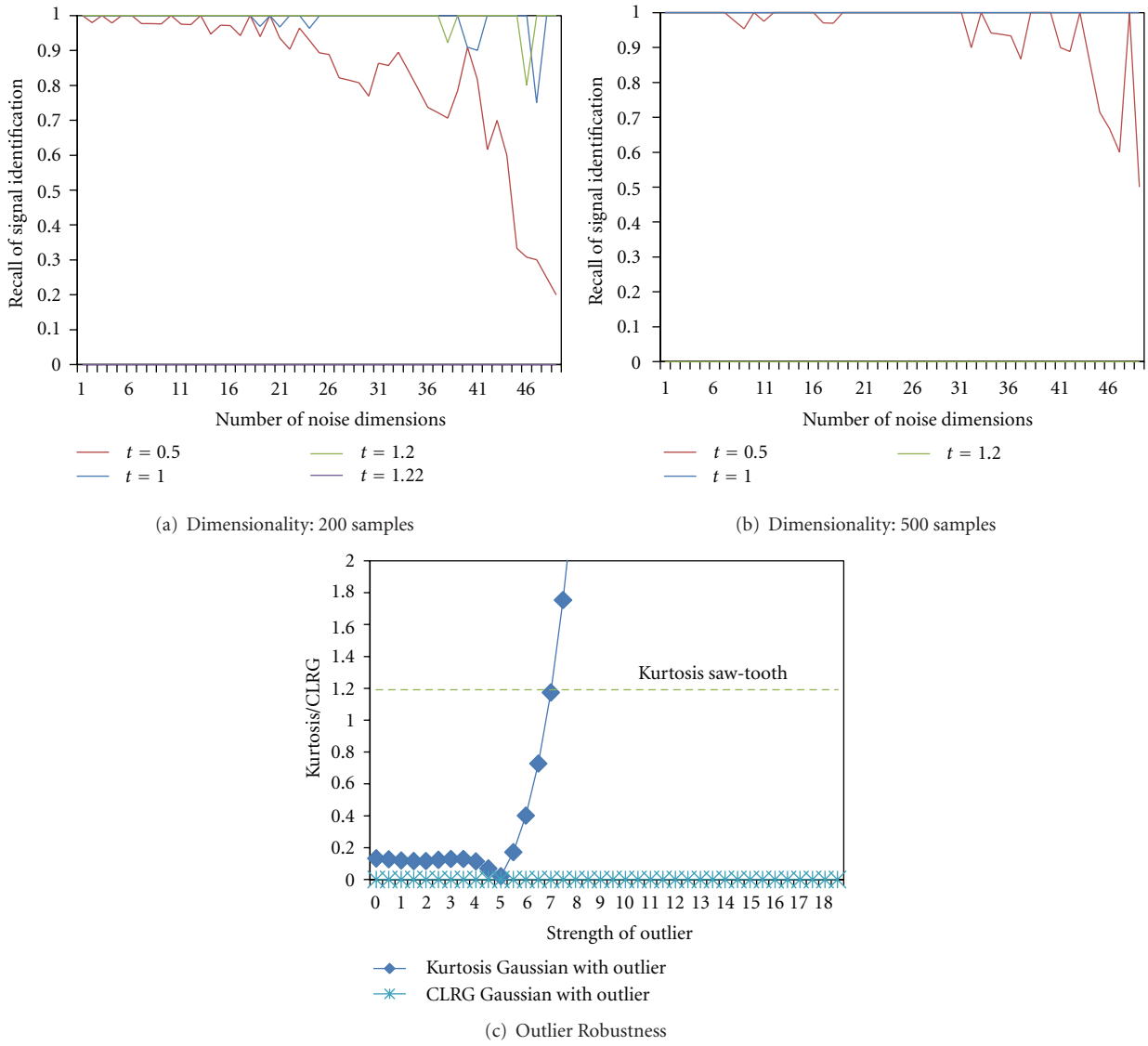
These steps starting from evaluation (22) are repeated until convergence.

## 4. Experiments

This section contains an extensive experimental evaluation. We start by a proof of concept demonstrating the benefits of information-theoretic model selection for ICA over established model selection criteria such as kurtosis in Section 4.1. Since in these experiments phi-transformed histograms and equidistant Gaussian Mixture Models perform very similar, for space limitations, we only show the results of phi-transformed histograms. In Section 4.2, we discuss the two possibilities of estimating the code length relative to Gaussianity.

### *4.1. Proof of Concept: Information-Theoretic Model Selection for ICA*

*4.1.1. Selection of the Relevant Dimensions.* Which ICs truly represent meaningful signals? Measures like kurtosis, skewness, and other approximations of neg-entropy are often used for selecting the relevant ICs but need to be suitably thresholded, which is a nontrivial task. Figures 3(a) and 3(b)




(a) Dimensionality: 200 samples

(b) Dimensionality: 500 samples

(c) Outlier Robustness

Figure 3: Comparison of CLRG to kurtosis for selection of relevant ICs from high-dimensional data (a)-(b) and for outlier-robust estimation of IC quality (c).

display the recall of signal identification for a dataset consisting of highly non-Gaussian saw-tooth signals and a varying number of noise dimensions for various thresholds of kurtosis. Kurtosis is measured as the absolute deviation from Gaussianity. The recall of signal identification is defined as the number of signals which have been correctly identified by the selection criterion divided by the overall number of signals. Figure 3(a) displays the results for various thresholds on a dataset with 200 samples. For this signal length, a threshold of $t = 1.2$ offers the best recall in signal identification for various numbers of noise dimensions. A slightly higher threshold of 1.22 leads to a complete break down in recall to 0, which implies that all noise signals are rated as non-Gaussian by kurtosis. For the dataset of 500 samples, however, $t = 1.0$ is a suitable threshold and for $t = 1.2$, we can observe a complete breakdown in recall. Even on these synthetic examples with a very clear distinction into highly non-Gaussian signals and Gaussian noise, the range for suitable thresholding is very narrow. Moreover, the threshold depends on the signal length and of course strongly on the type of the particular signal. A reasonable approach to select a suitable threshold is to try out a wide range of candidate thresholds and to select the threshold maximizing the area under ROC. For most of our example datasets with 200 samples, a threshold of $t = 1.2$ maximizes the area under ROC. For the datasets with 20 to 36 noise dimensions, this threshold yields a perfect result with an area under ROC of 1.0. For 500 samples, however, a lower threshold is preferable on most datasets. Supported by information theory, CLRG automatically identifies the relevant dimensions without requiring any parameters or thresholds. For all examples, CLRG identifies the relevant dimensions as those dimensions allowing data compression with a precision and a recall of 100%.

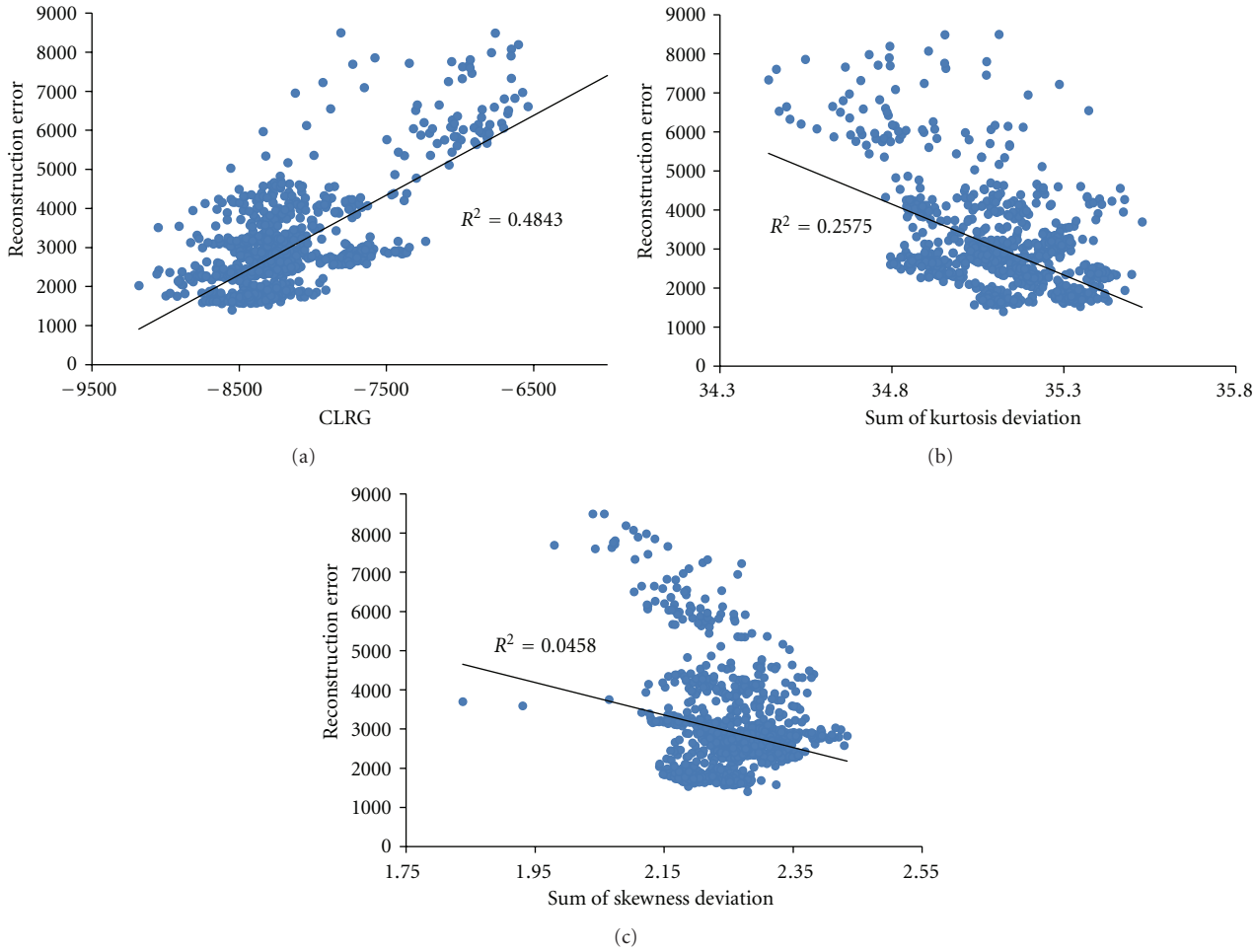



Figure 4: CLRG in comparison to kurtosis and skewness for assessing the quality of ICA-results. For 1,000 results obtained with FastIca on de-mixing speech signals, CLRG best correlates with the reconstruction error of the ICs.

*4.1.2. Stable Estimation of the IC Quality.* Commonly used approximations of neg-entropy are sensitive to single outliers. Outliers may cause an overestimation of the quality of the IC. CLRG is an outlier-robust measure for the interestingness of a signal. Figure 3(c) displays the influence of one single outlier on the kurtosis (displayed in terms of deviation from Gaussian) and CLRG of a Gaussian noise signal with 500 samples with respect to various outlier strengths (displayed in units of standard deviation). For reference, also the kurtosis of a highly non-Gaussian saw-tooth signal is displayed with a dotted line. Already for moderate outlier strength, the estimation of kurtosis becomes unstable. In case of a strong single outlier, kurtosis severely overestimates the interestingness of the signal. CLRG is not sensitive with respect to single outliers: even for strongest outliers, the noise signal is scored as not interesting with a CLRG of zero. For comparison, the saw-tooth curve allows an effective data compression with a CLRG of −553.

*4.1.3. Comparing ICA Results.* CLRG is a very general criterion for assessing the quality of ICA results which does not rely on any assumptions specific to certain algorithms. In this experiment, we compare CLRG to kurtosis and skewness on the benchmark dataset acspeec16 form ICALAB (http://www.bsp.brain.riken.go.jp/ICALAB/ICALABSignalProc/benchmarks/). This dataset consists of 16 speech signals which we mixed with a uniform random mixing matrix. Figure 4 displays 1,000 results of FastIca [2] generated with the nonlinearity tanh and different random starting conditions. For each result, we computed the reconstruction error as the sum of squared deviations of the ICs found by FastIca to the original source signals. For each IC, we used the best matching source signal (corrected for sign ambiguity) and summed up the squared deviations. Figure 4(a) shows that CLRG correlates best with the reconstruction error. In particular, ICA results with a low reconstruction error also allow effective data compression. For comparison, we computed the sum of kurtosis deviations and the sum of skewness deviations from Gaussianity. Kurtosis and even more skewness show only a slight correlation with the reconstruction error. As an example, Figure 5(a) shows the first extracted IC from the result best scored by CLRG and the corresponding IC (Figure 5(b)) from the result best scored by kurtosis. For each of the two ICs the scatter plots with

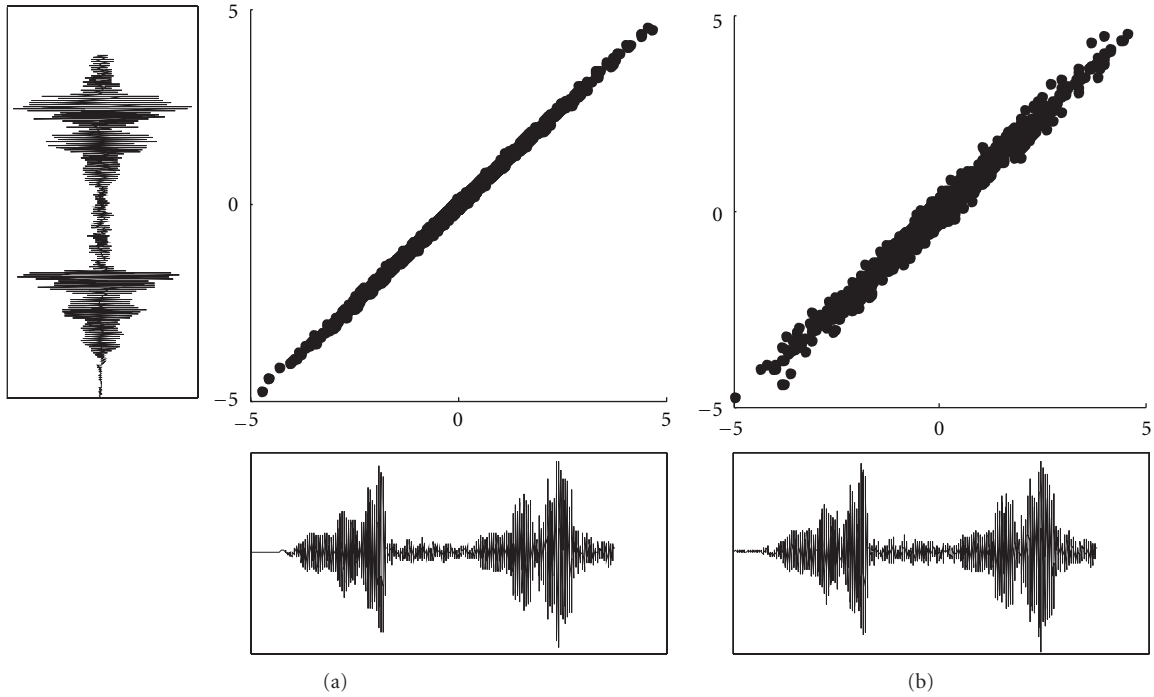



Figure 5: Reconstruction error of first IC from the result best scored by CLRG and matching IC from the result best scored by kurtosis.

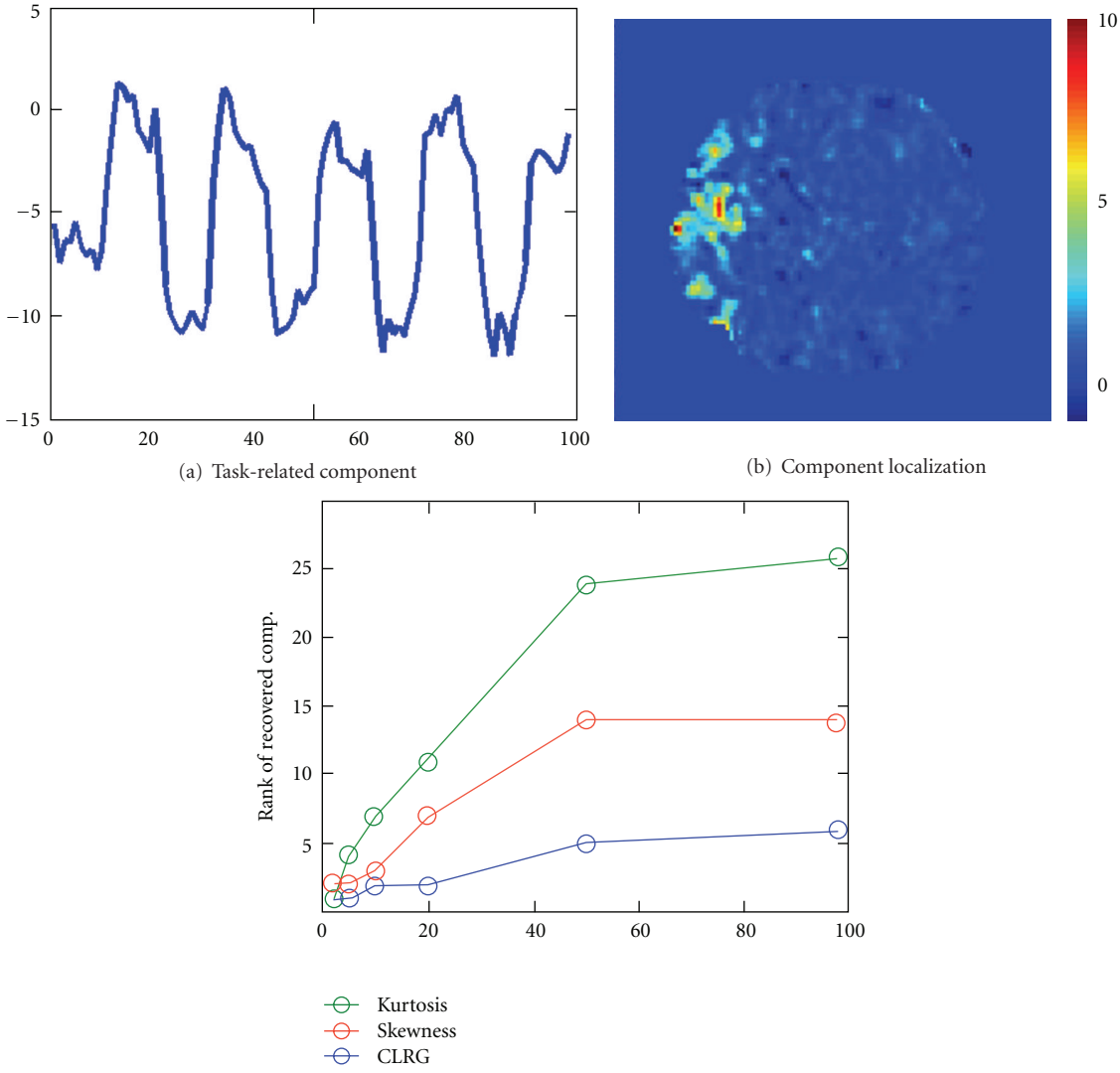
the original signal are displayed. Obviously, the left IC better matches the true signal than the right IC resulting in a lower reconstruction error.

*4.1.4. Selecting Relevant Components from fMRI Data.* Functional magnetic resonance imaging (fMRI) yields time series of 3-d volume images allowing to study the brain activity, usually while the subject is performing some task. In this experiment, a subject has been visually stimulated in a block-design by alternately displaying a checkerboard stimulus and a central fixation point on a dark background as control condition [21]. fMRI data with 98 images (TR/TE = 3000/60 msec) were acquired with five stimulation and rest periods having each a duration of 30 s. After standard preprocessing, the dimensionality has been reduced with PCA. FastIca has been applied to extract the task-related component. Figure 6(a) displays an example component with strong correlation to the experimental paradigm. This component is localized in the visual cortex which is responsible for processing photic stimuli, see Figure 6(b). We compared CLRG to kurtosis and skewness with respect to their scoring of the task-related component. In particular, we performed PCA reductions with varying dimensionality and identified the component with the strongest correlation to the stimulus protocol. Figure 6(c) shows that CLRG scores the task-related component much better than skewness and kurtosis. Regardless of the dimensionality, the task-related component is always among the top-ranked components by CLRG, in most cases among the top 3 to 5. By kurtosis and skewness, the interest of task-related component often rated close to the average.

*4.2. Discussion of CLRG Estimation Techniques.* Phi-transformed histograms and equidistant Gaussian mixture models represent different possibilities to estimate the code length relative to Gaussianity (CLRG). As elaborated in Section 3, to estimate the code length in bits, we need a probability density function (PDF) and the two variants differ in the way the PDF is defined. The major benefit of equidistant Gaussian mixture models over Phi-transformed histograms is that the PDF is defined by a continuous function which tends to represent some signals better than phi-transformed histograms. A better representation of the non-Gaussian characteristics of a signal results in more effective data compression expressed by a lower CLRG.

Figure 7 provides a comparison of phi-transformed histograms and equidistant Gaussian mixture models (eGMMs) regarding the CLRG estimated for the 16 signals of the aspeech16 dataset. For most signals, the CLRG estimated by both variants is very similar, for example, signals number 1 to 3, 10, and 16. Eight signals can be most effectively compressed using phi-transformed histograms, most evidently signals number 11 to 13. The other eight signals can be most effectively compressed using eGMM. In average on the aspeech16 dataset, the average CLRG 6,028 bits for phi-transformed histograms, 6,148 bits for eGMM.

We found similar results on other benchmark datasets also available at the ICALAB website: the 19 signals of the eeg19 dataset tend to be better represented by eGMM with an average CLRG of 17,691 (11 signals best represented by eGMM) followed by phi-transformed histograms with an average CLRG of 17,807 (8 signals best represented by phi-transformed).




(a) Task-related component

(b) Component localization

(c) Varing dim. reduction

Figure 6: fMRI experiment: (a) Task-related IC extracted by FastIca from a fMRI experiment where the subject performed a visual task while in the scanner. (b) Color-coded spatial activation pattern of this IC in an axial brain slice. (c) The rank of this IC according to CLRG and the comparison methods for varying dimensionality reduction. This interesting IC is always identified among the top-ranked components by CLRG.

Also, the abio7 data tends to be better represented by eGMM with an average CLRG of 6,480 (5 out of 7 signals best represented by GMM). Phi-transformed histograms perform with an average CLRG of 7,023 (2 best represented signals).

To summarize, we found only minor differences in performance among the two techniques estimating CLRG. Whenever a continuous representation of the PDF is required, the eGMM techniques should be preferred. A continuous representation allows, for example, incremental assessment of streaming signals. In this case, the CLRG can be reestimated periodically when enough novel data points have arrived from the stream.

## 5. Conclusion

In this paper, we introduced CLRG (code length relative to gaussianity) as an information-theoretic measure to evaluate the quality of single independent components as well as complete ICA results. Our experiments demonstrated that CLRG is an attractive complement to existing measures for non-Gaussianity, for example, kurtosis and skewness for the following reasons: relating the relevance of an IC to its usefulness for data compression, CLRG identifies the most relevant ICs in a dataset without requiring any parameters or thresholds. Moreover, CLRG is less sensitive to outliers

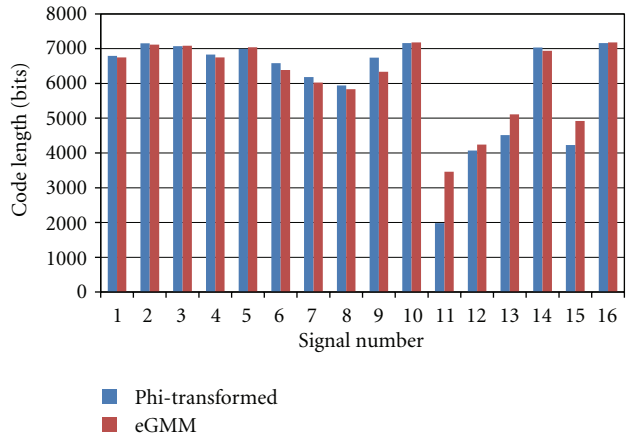



Figure 7: Comparison of CLRG estimation techniques regarding the compression of speech signals.

than comparison measures. On fMRI data, CLRG clearly outperforms the comparison techniques in identifying the relevant task-specific components.

The basic idea that a good model provides efficient data compression is very general. Therefore, not only different ICs and ICA results obtained by different algorithms can be unbiasedly compared. Given a dataset, we can also compare the quality completely different models, for example, obtained by ICA, PCA, and projection pursuit. Moreover, it might lead to the best data compression to apply different models to different subsets of the dimensions as well as different subsets of the data objects. In our ongoing and future work, we will extend CLRG to support various models and will explore algorithms for finding subsets of objects and dimensions which can be effectively compressed together.

## Acknowledgment

Claudia Plant is supported by the Alexander von Humboldt Foundation.

## References

[1] O. Sporns, "The human connectome: a complex network," *Annals of the New York Academy of Sciences*, vol. 1224, no. 1, pp. 109–125, 2011.

[2] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.

[4] P. M. Rasmussen, M. Mørup, L. K. Hansen, and S. M. Arnfred, "Model order estimation for independent component analysis of epoched EEG signals," in *Proceedings of the 1st International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS '08)*, pp. 3–10, January 2008.

[5] C. J. James and C. W. Hesse, "Independent component analysis for biomedical signals," *Physiological Measurement*, vol. 26, no. 1, pp. R15–R39, 2005.

[6] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[7] J. Himberg and A. Hyvärinen, "Icasso: software for investigating the reliability of ica estimates by clustering and visualization," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing (NNSP '03)*, pp. 259–268, 2003.

[8] F. Meinecke, A. Ziehe, M. Kawanabe, and K. R. Müller, "A resampling approach to estimate the stability of one-dimensional or multidimensional independent components," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 12, pp. 1514–1525, 2002.

[9] G. Qian, "Computing minimum description length for robust linear regression model selection," in *Proceedings of the Pacific Symposium on Biocomputing*, pp. 314–325, 1999.

[10] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Proceedings of the Asian Conference on Computer Vision (ACCV '09)*, vol. 5994 of *Lecture Notes in Computer Science*, pp. 135–146, 2009.

[11] T. Wekel and O. Hellwich, "Selection of an optimal polyhedral surface model using the minimum description length principle," in *Proceedings of the 32nd Symposium of the German Association for Pattern Recognition (DAGM '10)*, vol. 6376 of *Lecture Notes in Computer Science*, pp. 553–562, 2010.

[12] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 687–696, August 2007.

[13] D. Pelleg and A. W. Moore, "X-means: extending k-means with efficient estimation of the number of clusters," in *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pp. 727–734, 2000.

[14] C. Böhm, C. Faloutsos, J. Y. Pan, and C. Plant, "Robust information-theoretic clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 65–75, August 2006.

[15] C. Böhm, C. Faloutsos, and C. Plant, "Outlier-robust clustering using independent components," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 185–198, June 2008.

[16] C. Böhm, K. Haegler, N. S. Müller, and C. Plant, "CoCo: coding cost for parameter-free outlier detection," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 149–157, July 2009.

[17] P. Gruber, F. Theis, A. Tome, and E. Lang, "Automatic denoising using local independent component analysis," in *Proceedings of the 4th International ICSC Symposium on Engineering of Intelligent Systems (EIS '04)*, 2004.

[18] A. Barron, J. Rissanen, and B. Yu, "The Minimum Description Length Principle in Coding and Modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.

[19] T. C. M. Lee, "Regression spline smoothing using the minimum description length principle," *Statistics and Probability Letters*, vol. 48, no. 1, pp. 71–82, 2000.

[20] T. W. Lee, M. S. Lewicki, and T. J. Sejnowski, "ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1078–1089, 2000.

[21] A. Wismüller, O. Lange, D. R. Dersch et al., "Cluster analysis of biomedical image time-series," *International Journal of Computer Vision*, vol. 46, no. 2, pp. 103–128, 2002.