PCA GAUSSIANIZATION FOR IMAGE PROCESSING

Valero Laparra, Gustavo Camps-Valls and Jesús Malo

Image Processing Laboratory (IPL), Universitat de València
Catedrático A. Escardino - 46980 Paterna, València, Spain
{lapeva,gcamps,jmalo}@uv.es

ABSTRACT

The estimation of high-dimensional probability density functions (PDFs) is not an easy task for many image processing applications. The *linear* models assumed by widely used transforms are often quite restrictive to describe the PDF of natural images. In fact, additional non-linear processing is needed to overcome the limitations of the model. On the contrary, the class of techniques collectively known as *projection pursuit*, which solve the high-dimensional problem by sequential univariate solutions, may be applied to very general PDFs (e.g. iterative Gaussianization procedures). However, the associated computational cost has prevented their extensive use in image processing.

In this work, we propose a fast alternative to iterative Gaussianization methods that makes it suitable for image processing while ensuring its theoretical convergence. Method performance is successfully illustrated in image synthesis and classification problems.

Index Terms— Gaussianization, PCA, density estimation, image synthesis, one-class image classification.

1. INTRODUCTION

Many image processing applications such as coding, restoration, classification or synthesis greatly depend on an appropriate description of the PDF of the signal. However, density estimation is a challenging problem when dealing with high-dimensional signals because direct sampling of the input space is not an easy task due to the curse of dimensionality.

The aim of image representations (or transforms) is to include the properties of the signal in the transform parameters. However, the most popular representations rely on linear models that are too restrictive to describe natural images globally. For instance, PCA and local DCT assume a Gaussian source, while linear ICA and wavelets assume that images come from the *linear* combination of independent sources. These assumptions are not completely correct: for instance, a usual combination rule in natural scenes such as occlusion is intrinsically non-linear. This implies that residual relations among coefficients still remain after any linear transform. The unsuitability of linear transforms to encompass the complexity of natural images implies that a number of tricks have to be added after the linear transform in order to describe the remaining relations. Examples of successful characterization of post-transforms relations include texture synthesis [1], image coding [2, 3], or image denoising [4].

On the contrary, the class of techniques collectively known as *projection pursuit* [5, 6] may be applied to very general PDFs. These techniques solve the high-dimensional density estimation problem by successive univariate solutions thus circumventing the curse of dimensionality. For instance, the Gaussianization procedure proposed in [7] performs a series of linear ICA transforms followed by marginal Gaussianization in every transformed dimension. We will refer to this particular projection pursuit technique as GICA. Since convergence is guaranteed, after an *appropriate* number of iterations, any arbitrary PDF can be turned into a unit variance Gaussian, and thus (unlike linear transforms) complete independence among coefficients is achieved. The richness of the PDF under consideration is captured by the series of ICA transforms and the corresponding marginal non-linearities.

The weakness of general projection pursuit techniques, and also of GICA, is their computational cost. Note that, in this case, ICA is performed in each iteration: robust ICA algorithms such as RADICAL [8] lead to extremely slow convergence while convenient alternatives such as FastICA [9] may not converge. This explains why, so far, GICA has been applied just to low-dimensional (audio) signals [10, 11].

These problems could be alleviated by the recently proposed single-step (non-iterative) Gaussianization transforms [12, 13]. Unfortunately, these single step procedures are *also* restricted to particular PDF classes: (1) PDFs defined in convex domains so that the final Gaussian can be achieved by marginal Gaussianization of every dimension in the appropriate axes [12], or (2) elliptically symmetric PDFs so that the final Gaussian can be achieved by equalizing the length (norm) of the whitened samples [13]. In the case of images, the elliptical symmetry, and consequently convex domain, is true for small image patches (e.g. 10×10 pixels), but does not hold for bigger neighborhoods [13]. According to this, a general (yet computationally affordable) PDF estimation technique suited to image processing applications is not available yet.

In this work, we propose a fast alternative to GICA [7] that makes it suitable for image processing applications. In each iteration, we use the standard PCA as an alternative to linear

This work was partially supported by projects CICYT-FEDER TEC2006-13845, AYA2008-05965-C04-03 and CSD2007-00018, and grant TACATAC.

ICA, thus obtaining the desired Gaussianization through iterated PCA and marginal Gaussianization (GPCA). Here, we show that using an orthogonal linear transform in the iterative procedure does not change the theoretical convergence nor the convergence rate in practice. As a result, a much faster procedure is obtained while keeping the appealing properties of the original method.

The paper is outlined as follows. Section 2 reviews the original GICA formulation. Section 3 presents the proposed GPCA method, demonstrates its theoretical convergence, and shows that, in practice, GPCA converges to GICA-like solutions in a fraction of the time. Section 4 illustrates the potential of Gaussianization in image processing by using the proposed GPCA in synthesis and classification problems. Finally, Section 5 draws the conclusions of the work.

2. ICA GAUSSIANIZATION (GICA)

Given a d-dimensional random variable $\mathbf{x}^{(0)}$, in each iteration k, the ICA-based Gaussianization (GICA) performs:

$$\mathbf{x}_{(k+1)} = \Psi_{(k)}(\mathbf{A}_{(k)}\mathbf{x}_{(k)}), \tag{1}$$

where $\mathbf{A}$ is a linear transform, and Ψ is the marginal Gaussianization of each dimension of $\mathbf{A}\mathbf{x}$.

It can be shown that by choosing the matrices $\mathbf{A}_{(k)}$ that minimize the mutual information of transformed data, the series of the corresponding PDFs, $p(\mathbf{x}_{(k)})$, converges to a multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{I})$ [7]. Since ICA techniques fulfil the above requirement by definition, these are used to compute $\mathbf{A}_{(k)}$. Note that, in this particular projection pursuit method, the general idea of seeking for *interesting* projections reduces to looking for the most independent projected features in each iteration.

The interesting property of GICA is that any dataset can be turned into a multivariate Gaussian through an invertible and differentiable transform. Being a Gaussian distribution (fully independent features) implies that the complexity of the original PDF is encoded in the transform. Transform invertibility allows us to achieve solutions in the original domain while operating in a well-characterized (Gaussian) domain. Finally, as the transform is differentiable, one can estimate the PDF in the original domain from the Jacobian in each point.

The main problem of GICA is its computational cost, as it relies on performing ICA at every iteration. ICA has no closed-form solution and iterative methods must be deployed. Robust ICA algorithms, such as RADICAL [8], are extremely slow while convenient alternatives, such as FastICA [9], may not converge in all cases. A second and critical problem is that, surprisingly, no practical criterion to stop the iterative procedure was proposed in [7]. However, note that after a number of iterations, no significant gain may be achieved in terms of independence of the transformed features.

3. PCA GAUSSIANIZATION (GPCA)

Here, we propose to solve the aforementioned problems of GICA by replacing linear ICA transforms $\mathbf{A}_{(k)}$ with a series of orthogonal transforms $\mathbf{B}_{(k)}$ obtained through linear PCA, referred to as GPCA. Unlike ICA, using PCA ensures a closed-form stable and unique solution and the computational burden is dramatically reduced. While this may seem a naïve solution, some non-trivial questions arise:

- Is convergence of the new algorithm guaranteed?
- Do GICA and GPCA solutions differ?

In the following subsections we address these questions both theoretically and experimentally.

3.1. Convergence of GPCA

The series of PDFs corresponding to $\mathbf{x}_{(k)}$ following (1) and fulfiling

$$\lim_{k\to\infty}\left(\sup_{\|\alpha\|_2=1} J(\alpha^\top \mathbf{x}_{(k)})\right) = 0 \tag{2}$$

converges to a normal distribution $\mathcal{N}(0, \mathbf{I})$ [6], where $J(\cdot)$ is the negentropy (i.e. the Kullback-Leibler divergence between a random variable and a normal), and α represents any possible orthogonal projection of the data.
It is easy to see that

$$J^* = \sup_{\|\alpha\|_2=1} J(\alpha^\top \mathbf{x}_{(k)}) \leq I(\mathbf{x}_{(k)}) - \inf_{\mathbf{U}} I(\mathbf{U}^\top \mathbf{x}_{(k)}), \tag{3}$$

where $\mathbf{U}$ is any possible orthogonal matrix and $I(\cdot)$ is the mutual information. Note that the infimum term in (3) will be smaller for non-orthogonal matrices $\mathbf{A}$, and thus

$$J^* \leq I(\mathbf{x}_{(k)}) - \inf_{\mathbf{A}} I(\mathbf{A}\mathbf{x}_{(k)}). \tag{4}$$

ICA computes matrices $\mathbf{A}_{(k)}$ such that mutual information is minimized. In addition, it can be shown that the negentropy reduction in each iteration $\Delta^{(k)}_{\text{ICA}}$ equals the right hand side term in (4), and its limit is zero. Hence, since J^* is bounded by $\Delta^{(k)}_{\text{ICA}}$, $p(\mathbf{x}_{(k)}) \to \mathcal{N}(0, \mathbf{I})$.

When using PCA instead of ICA, it is not possible to obtain (4) since PCA matrices $\mathbf{B}_{(k)}$ do not necessarily minimize mutual information among all possible orthogonal transforms $\mathbf{U}$. However, one can always define an upper bound for J^* by using an $\varepsilon > 0$ such that,

$$J^* \leq I(\mathbf{x}_{(k)}) - \inf_{\mathbf{U}} I(\mathbf{U}^\top \mathbf{x}_{(k)}) \leq \frac{1}{\varepsilon}\left(I(\mathbf{x}_{(k)}) - I(\mathbf{B}^\top_{(k)}\mathbf{x}_{(k)})\right)$$

provided that $I(\mathbf{x}_{(k)}) - I(\mathbf{B}^\top_{(k)}\mathbf{x}_{(k)}) \geq 0$. If, in addition, this upper bound tends to zero, convergence of GPCA is guaranteed. In the following, these conditions are demonstrated.

Upper bound is positive. Negentropy reduction in each iteration when using PCA reduces to $\Delta^{(k)}_{\text{PCA}} = I(\mathbf{x}_{(k)}) - I(\mathbf{B}^\top_{(k)}\mathbf{x}_{(k)})$. The mutual information can be decomposed into second and higher order terms:

$$I(\mathbf{x}_{(k)}) = \sum_{i=1}^{d} \log(\mathbf{C}_{ii}) - \log(|\mathbf{C}|) + J(\mathbf{x}_{(k)}) - J_M(\mathbf{x}_{(k)}),$$

where $\mathbf{C}$ is the covariance matrix of $\mathbf{x}_{(k)}$, and $J_M(\cdot)$ is the marginal negentropy defined as the sum of univariate negentropies. Since $\mathbf{x}_{(k)}$ is marginally gaussianized, $\mathbf{C}_{ii} = 1\ \forall i$, and therefore the first and last terms vanish. Also, note that as the sum of the eigenvalues is $\sum_{i=1}^{d} \lambda_i = tr(\mathbf{C}) = d$, $\Pi_i \lambda_i \leq 1$, then the second term $\log(|\mathbf{C}|) = \log(\Pi_i \lambda_i) \leq 0$. After PCA, the covariance is diagonal $\mathbf{\Lambda}$, and the mutual information is

$$I(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) = \log(\Pi_i \lambda_i) - \log(|\mathbf{\Lambda}|) + J(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) - J_M(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}).$$

Since $J(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) = J(\mathbf{x}_{(k)})$, $|\mathbf{C}| = |\mathbf{\Lambda}|$, and given that $J_M(\cdot) \geq 0$ by definition, then

$$\Delta_{\text{PCA}}^{(k)} = J_M(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) - \log(\Pi_i \lambda_i) \geq 0$$

Upper bound tends to zero. As $I(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) \geq I(\mathbf{A}_{(k)}\mathbf{x}_{(k)})$, then $\Delta_{\text{PCA}}^{(k)} \leq \Delta_{\text{ICA}}^{(k)}$. Since $\Delta_{\text{ICA}}^{(k)}$ tends to zero, and $\Delta_{\text{PCA}}^{(k)} \geq 0$, as demonstrated before, then $\Delta_{\text{PCA}}^{(k)}$ tends to zero.

3.2. Stopping Criterion

The previous theoretical convergence limit does not provide a practical criterion to stop the iteration. Note that one should stop the series of transforms when the reduction in negentropy (distance to a Gaussian) is small enough. On the one hand, when approaching the theoretical limit (infinite iterations), the reduction is not zero but the different solutions are quite similar. Therefore, in practice, a much lower number of iterations is needed. On the other hand, another practical issue concerns cross-validation: in order to avoid over-fitting to a particular data set, it is convenient to train the transform with a representative data subset and evaluate the information measurement in an independent yet representative test subset. Therefore, we propose to stop the iteration when the information measure is minimum in this test subset.

This criterion involves computing the negentropy reduction in the test subset at each iteration,

$$\Delta_{\text{PCA}}^{(k)} = \sum_{i=1}^{d} H(\mathbf{x}_{(k)}) - \sum_{i=1}^{d} H(\mathbf{B}_{(k)}^{\top}\mathbf{x}_{(k)}) + \mathbb{E}\left[\log(|\mathbf{B}_{(k)}^{\top}|)\right]$$

where only univariate (reliable) entropy measures, H, are needed since $|\mathbf{B}_{(k)}^{\top}| = 1$.

3.3. Performance of GPCA vs GICA

Here, we analyze two important characteristics of GPCA: the convergence rate and the computational cost. Figure 1 illustrates the performance of our method in a 2D highly non-Gaussian manifold. The proposed early-stopping criterion was applied: the transform was learned with $2/3$ of the data and validated in the rest. Figure 1 shows how at each iteration the (accumulated) redundancy reduction ΔI converges for both GICA[1] and GPCA, and our method achieves virtually the same results with a slightly higher number of iterations (37 vs 25). Note, however, that it does not imply a higher computational load, as PCA is much cheaper than ICA. This advantage is more relevant in higher dimensional problems. To assess this, we Gaussianized patches of different sizes from the image 'Barbara'. Results for both CPU time and the achieved ΔI are presented in Table 1. For similar ΔI reductions, more than an order of magnitude in computation time is gained by GPCA, e.g. when working with 8×8 patches GPCA takes about 4 minutes while GICA takes around 4 hours.

[1] We used the FastICA algorithm [9] to speed up learning.

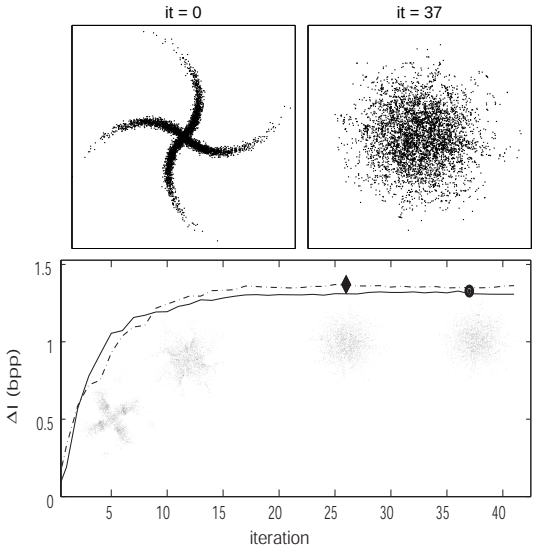

Fig. 1. Performance of GPCA in a toy example. Original and transformed data (top), and cumulative ΔI for each iteration for GPCA (solid) and GICA (dashed). Optimal iterations are highlighted. Inset scatter plots show the achieved GPCA solution at different iterations.

	GPCA		GICA	
dim	ΔI [bpp]	Time [s]	ΔI [bpp]	Time [s]
2×2	1.51	14	1.54	865
3×3	2.05	34	2.08	1236
4×4	2.29	63	2.38	2197
5×5	2.44	99	2.50	3727
6×6	2.56	141	2.60	6106
7×7	2.63	170	2.68	9329
8×8	2.69	233	2.69	15085

Table 1. Cumulative ΔI and CPU time for GICA and GPCA.

4. EXPERIMENTAL RESULTS

This section presents the capabilities of the proposed GPCA in two image processing applications: image synthesis and target detection in multispectral images.

Experiment 1: Image Synthesis. A nice property of GPCA is *invertibility*. In this experiment, we applied GPCA to describe the PDF of a database of 2400 faces [14]. Images cropped to 17×15 pixels were used to learn the transform that

'Gaussianizes' the data. Then, the inverse of the transform was applied to random samples generated from a multivariate Gaussian. This operation leads to new synthetic faces in the original space. Figure 2 shows real (top) and synthesized images (bottom), which are a realistic representation of the learned PDF. Finally, note that dimension $d = 17 \cdot 15 = 255$, is unaffordable for GICA (cf. Table 1).

Fig. 2. Original (top) and GPCA synthetic faces (bottom).

Experiment 2: Multisource one-class image classification. We stacked at a pixel level seven Landsat bands, two SAR backscattering intensities, and its interferometric coherence for *target detection* ($d = 10$). We compare the performance of the GPCA with the support vector domain descriptor (SVDD) since they are conceptually similar. On the one hand, GPCA learns the class of interest ('urban') in the scene. Then test samples are transformed and classified as *target* if they lie inside the sphere containing $1 - \nu$ fraction of the Gaussian distribution. On the other hand, SVDD finds a minimum volume sphere in a kernel feature space that contains $1 - \nu$ fraction of the training samples [15]. We used the RBF kernel for the SVDD whose width was varied in the range $\sigma \in [10^{-3}, 10^{3}]$. The fraction rejection parameter was varied in $\nu \in [0, 10^{-1}]$ for both methods. The best parameters were selected through 3-fold cross-validation in the training set (1500 pixels). The experiment was repeated for 20 different random realizations. The average overall accuracy (OA) and kappa statistic (κ) in the test set (40000 pixels) for the SVDD were OA = 87 ± 9 and $\kappa = 0.75 \pm 0.19$, while GPCA obtained OA = 91 ± 4 and $\kappa = 0.83 \pm 0.07$. Figure 3 shows the classification maps for a representative realization. Note that GPCA better rejects the 'non-urban' areas (in white).

5. CONCLUSIONS

We proposed a fast alternative to iterative Gaussianization methods that makes it suitable for image processing. The proposed GPCA consists of iteratively applying PCA and marginal Gaussianization to any original dataset, thus leading to a multivariate Gaussian. Theoretical convergence of the proposed method has been proved. It exhibits fast and stable convergence rates through a suitable early-stopping criterion. Finally, the computational cost is dramatically reduced compared to ICA-based Gaussianization methods. Method performance is successfully illustrated in image synthesis and classification problems.

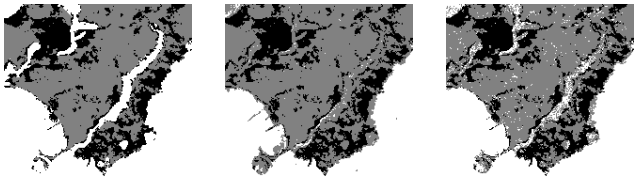

Fig. 3. Ground truth (left), SVDD (middle), GPCA (right). Black: unknown class, Gray: urban, White: non-urban.

6. REFERENCES

[1] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comp. Vis.*, vol. 40, no. 1, pp. 49–71, 2000.

[2] J. Malo, I. Epifanio, R. Navarro, and E. Simoncelli, "Nonlinear image representation for efficient perceptual coding," *IEEE Trans. Im. Proc.*, vol. 15, no. 1, pp. 68–80, 2006.

[3] G. Camps-Valls, J. Gutiérrez, G. Gómez, and J. Malo, "On the suitable domain for SVM training in image coding," *Journal of Machine Learning Research*, vol. 9, pp. 49–66, 2008.

[4] V. Laparra, J. Gutiérrez, G. Camps-Valls, and J. Malo., "Recovering wavelet relations using SVM for image denoising," *IEEE ICIP 08*, pp. 541–544, 2008.

[5] J.H. Friedman and J.W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comp.*, vol. C-23, no. 9, pp. 881–890, 1974.

[6] P. J. Huber, "Projection pursuit," *The Annals of Statistics*, vol. 13, no. 2, pp. 435–475, 1985.

[7] S.S Chen and R.A. Gopinath, "Gaussianization," in *NIPS*, 2000, pp. 423–429.

[8] E.G. Learned and J.W. Fisher, "ICA using spacings estimates of entropy," *J. Mach.Learn.Res.*, vol. 4, pp. 1271–1295, 2003.

[9] A. Hyvärinen, "Fast and robust fixed-point algorithms for ICA," *IEEE Trans. Neur. Net.*, vol. 10, pp. 626–634, 1999.

[10] B. Xiang, U.V. Chaudhari, G.N. Ramaswamy, and R.A. Gopinath, "Short-time gaussianization for robust speaker verification," in *IEEE ICASSP 02'*, Orlando, Florida, 2002.

[11] K. Zhang and L.W. Chan, "Extended gaussianization method for blind separation of post-nonlinear mixtures," *Neur. Comp.*, vol. 17, no. 2, pp. 425–452, 2005.

[12] D. Erdogmus, R. Jenssen, Y. Rao, and J. Principe, "Gaussianization: An efficient multivariate density estimation technique for statistical signal processing," *J. VLSI Sig. Proc.*, vol. 45(17), pp. 67–83, 2006.

[13] S Lyu and E P Simoncelli, "Nonlinear extraction of 'independent components' of natural images using Radial Gaussianization," *Neur. Comp.*, 2009, Accepted for publ., 10/08.

[14] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[15] D. Tax and R.P.W. Duin, "Support vector domain description," *Patt. Recogn. Lett.*, vol. 20, pp. 1191–1199, 1999.