NONPARAMETRIC ESTIMATION AND TRACKING OF THE MIXING MATRIX FOR UNDERDETERMINED BLIND SOURCE SEPARATION

Deniz Erdoğmuş,[*] *Luis Vielva,*[★] *José C. Príncipe*[*]

[*] Computational NeuroEngineering Laboratory, University of Florida, Gainesville, FL
[★] Communications Engineering Laboratory, Universidad de Cantabria, Spain
E-mail: {deniz, principe}@cnel.ufl.edu, luis@dicom.unican.es

ABSTRACT

Blind source separation deals with the problem of estimating n source signals from m measurements, which are generated through an unknown mixing process. In the underdetermined linear case, where the number of measurements is smaller than the number of sources, the solution can be obtained in three stages: find a sparse representation domain for the signals, find the mixing matrix, and estimate the sources using the previous knowledge. This paper addresses the second stage. A non-parametric maximum-likelihood approach based on Parzen windowing is presented. It is shown that the peak points in the probability distribution of measurements directions correspond to the directions of the column vectors of the mixing matrix. An algorithm to estimate the column vectors in the static case, and to track the column vectors in the dynamic case is presented. The tracking capability of the algorithm is determined and, using a simple wave propagation model, corresponding limitations on the speeds of mobile sources are derived.

1. INTRODUCTION

The blind source separation (BSS) problem is defined as the identification of the n unknown statistically independent source signals from m observations, which are generated by an unknown mixing procedure. In the noise-free linear instantaneous underdetermined case, the number of observations is smaller than the number of sources and one can write the observation vector as a linear transformation on the source vector as

$$\mathbf{A}\mathbf{s} = \mathbf{x}.$$

If the signals are sufficiently sparse, or if a suitable sparse transformation can be applied, the sources can be estimated from the measurements once the mixing matrix is known [1], [2]. There have been different approaches taken towards estimating the mixing matrix. Lin et. al use competitive learning in a feature extraction framework [3]. Bofill and Zibulevsky employ a potential function based clustering approach [4]. On the other side, Wu uses an eigenspread estimation to decide when only one source is active, and uses this information to find the columns of the mixing matrix [5]. In this paper we use a non-parametric maximum-likelihood approach, based on Parzen windowing. In this method probability distribution of sample directions is non-parametrically estimated and the peak points are shown to correspond to the directions that define the column vectors of the mixing matrix. Two different versions of the training algorithm are provided, one for both the static and one for the dynamic case. The limitations on the tracking capability of the algorithm are determined and, using a simple wave propagation model, corresponding limitations on the speeds of mobile sources are derived.

2. ESTIMATION OF THE MIXING MATRIX

Suppose the following sparse model describes the source distributions,

$$p_{S_j}(s_j) = p_j\,\delta(s_j) + (1-p_j) f_{S_j}(s_j), \quad j = 1, \ldots, n, \quad (1)$$

where p_j is the sparsity factor for source j, and f_{S_j} is the density when the source is active. In addition, it is assumed that the sources are zero-mean. If the sparsity factor is large, quite often a number of sources will be silent at a given time instant, and occasionally only one source will produce a nonzero signal. When this is the case, the measurement at that instant will be collinear with the corresponding column of the mixing matrix. The scatter-plot of one such mixing process with two measurements and three sources, where all the sources have a fixed sparsity factor of 0.2 is presented in figure 1a. Notice that when the histogram of the angle of samples is considered, as shown in figure 1b, the three directions designated by the columns of the mixing matrix are clearly identified. It is remarkable that even with sparsity factors of as low as 10%, it is possible to distinguish from the histogram of the angle of samples the columns of $\mathbf{A}$. However, the resolution of the histogram-based estimation of the column vectors is directly determined by the bin length that is assumed in evaluating the histogram. To over-

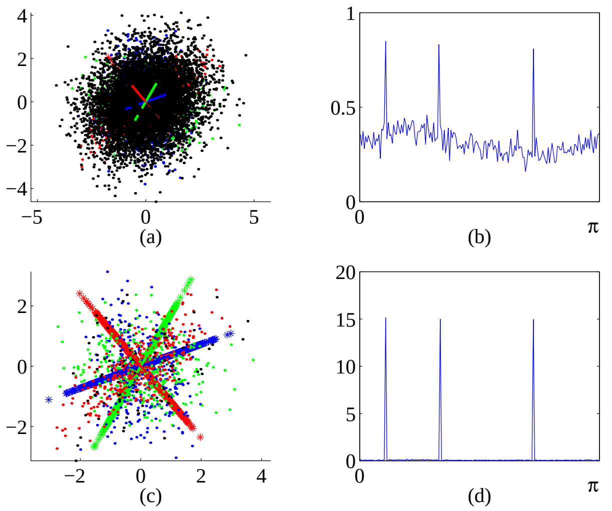

Fig. 1. Scatter plot of measurements and histogram of angles for sparsity factors 0.1 —(a) and (b)— and 0.8 —(c) and (d).

come this problem, we propose the use of Parzen windowing [6] to estimate the probability density of the angle, and then pick the n largest peaks of this distribution as the estimates for the directions of the n columns of the mixing matrix $\mathbf{A}$. In this paper, we restrict ourselves to the three source-two measurement case in order to be able to present better visualizations of the concepts. In that case, the directions of the columns of the mixing matrix can be identified with only a single angle, thus it is a one-dimensional random variable.

The Parzen window density estimation for the angle given the samples and a kernel function $\kappa_\sigma(\cdot)$ is given by

$$p(\theta) = \frac{1}{N}\sum_{i=1}^{N}\kappa_\sigma(\theta - \theta_i), \qquad (2)$$

where the samples of the angle are evaluated, using the nonzero measurement vectors, from

$$\theta_i = \arctan\frac{x_2}{x_1}.$$

The zero measurements are simply omitted, as they have no well-defined angle. Once this density estimation is obtained, standard optimization methods to find the angles corresponding to the peaks of the density function can be employed. In this study, we utilize the steepest ascent algorithm to achieve this objective. There are two cases of interest that require different approaches when the steepest ascent is to be used in this problem. These are the static case, where the mixing matrix is constant at all times, and the dynamic case, where the mixing matrix is time varying, but the mixture is still instantaneous. In the following sections, we will provide algorithms to achieve optimal solutions for both cases.

3. STATIC CASE

In the static case, the mixing matrix is assumed to be constant; therefore, all the measurement samples can be used in the density estimation for the angle in a batch-learning scheme. Since we are looking for the largest three peaks (due to three sources) of the estimated density, and we are going to use steepest ascent, our initial estimates must be in the domain of attraction of those solutions that we seek. For this, the direction estimates obtained from the histogram method are used as initial conditions to the steepest ascent algorithm. For example, by using 180 bins in the interval $[0, \pi]$ we can obtain initial estimates that are closer than one degree to the solutions. Note that it is sufficient to consider the angles in this interval only, since a sign ambiguity is acceptable in BSS. Furthermore, we can assume that the columns of $\mathbf{A}$ are unit length, since this corresponds to an ambiguity in the scaling factor, which is also acceptable in BSS.

Once the initial estimates are obtained from the histogram method, the following gradient expression of the 'cost function' in (2) is used to refine the estimates until convergence to the maximum is achieved

$$\nabla_\theta p(\theta) = \frac{1}{N}\sum_{i=1}^{N}\nabla\kappa_\sigma(\theta - \theta_i).$$

This procedure is repeated for the three initial conditions provided by the histogram. Since Parzen windowing provides a continuous estimate of the density function of interest, in theory it is possible to achieve a very high resolution, provided that the kernel size is chosen sufficiently small so that there are no artifact peaks; whereas in the histogram, in order to increase resolution, more bins are necessary.

The choice of the kernel size in Parzen windowing is crucial to an accurate estimation of the correct directions of the columns. In figure 2, the MSE in the estimation of the actual directions (in radians squared) versus the kernel size for Gaussian kernels is shown for sparsity factors of 0.2, 0.5, and 0.8. We have defined the MSE as

$$\frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{3}\frac{1}{3}(\theta_j - \hat{\theta}_j)^2,$$

where M is the number of simulations used in the Montecarlo method. It is clear from this plot that, given a step size for steepest ascent, there exists an optimal kernel size for each sparsity factor. However, since there is no reference to compare estimates in practice, it is safer to use a large kernel size as the estimation MSE does not increase as fast as it does when a smaller kernel size is used.

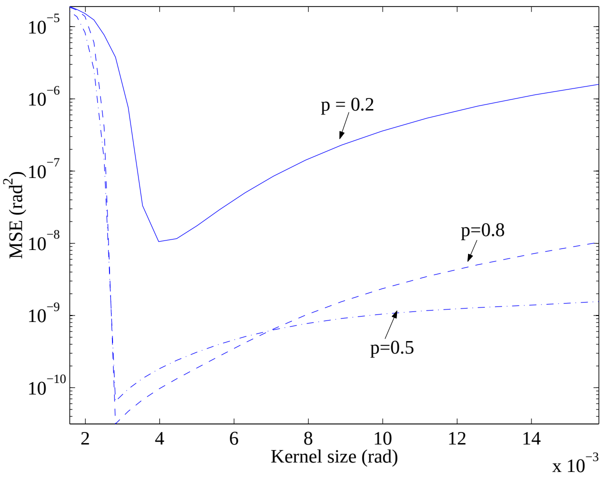

Fig. 2. MSE in angle estimation vs kernel size for different sparsity factors.

It is also of theoretical interest to investigate the behavior of MSE of estimation as a function of the sparsity rate. One would expect to observe an increasing performance in estimation as the sparsity increases, since there will be more and more samples that are perfectly aligned with the columns compared to the outliers that are generated as a result of more than one active source.

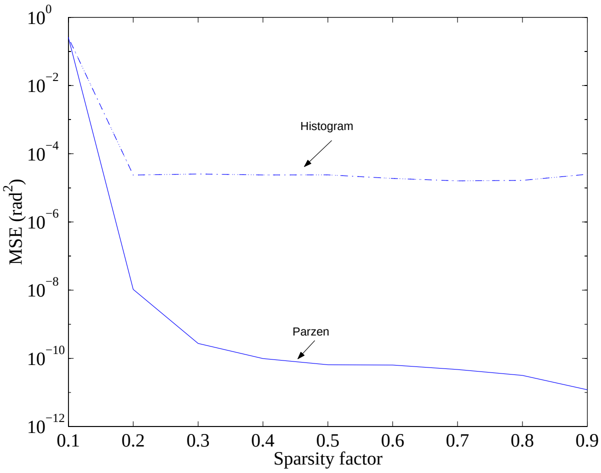

Fig. 3. MSE in angle estimation vs sparsity factor for Parzen window (using optimal kernel size) and histogram.

Figure 3 shows that, just as expected, when the sparsity rate increases the MSE decreases. Another very important conclusion drawn from this plot is that refining the estimates with the Parzen windowing method remarkably improves upon the initial estimates given by the histogram. Note that, since a steepest ascent algorithm with a fixed step size is utilized in training, the actual maxima are not exactly obtained. A portion of the MSE in the Parzen window method is thus due to this slight misadjustment from the optimal values. In contrast, a greater portion of the MSE in the histogram estimates is due to the finite bin length. Assuming a uniform distribution in each bin of length one degree, the associated variance would be on the order of 10^{-5} rad^2 . We observe that the results in figure 3 are in conformity with this expectation.

4. DYNAMIC CASE

In the dynamic case, we assume that the mixing matrix is time varying, although the mixing procedure is still linear and instantaneous. This situation may occur, for example, when there are fixed microphones in a room and the speakers are moving around, so that the attenuation experienced by the speech signal until it reaches the microphone varies with time. Assuming no echo and reverberation in the environment, and also assuming that the speech from a speaker arrives at all microphones at the same instant, this time-varying instantaneous mixture model is sufficient.

In order to track the columns of a changing matrix, the training algorithm presented in the static case must be modified slightly. Since the algorithm will try to track the directions of the columns blindly, a sufficiently accurate initial estimate is crucial. The static Parzen windowing method can be used to achieve this initialization assuming that the columns are rotating 'slowly'. In order to initialize the angle estimations, a number of samples must be collected first. If the time variation of the columns is slow, then when the static training algorithm is applied to this data set, one obtains a sufficiently accurate initial estimate of the angles. The number of samples required for this initialization procedure can be in the order of hundreds or smaller. Once the initialization is achieved, a modified version of the steepest ascent algorithm will be applied to adapt the angle estimates on-line on a sample-by-sample basis. As the adaptation criterion, there exist alternatives. One of them is to use a fixed-length window of samples extending back in time and use these samples to update the density estimate with every new sample. Then, the solutions that maximize the estimated density can be obtained using steepest ascent. A second approach, the one we will adopt, is to use a forgetting factor approach and to estimate the new density using a linear combination of the estimate from the previous sample and the kernel evaluated at the current sample. In this case, the cost function, i.e. the density estimate, at time instant k reads as

$$p_k(\theta) = \alpha p_{k-1}(\theta) + (1-\alpha)\kappa_\sigma(\theta-\theta_k),$$

where α is the forgetting factor. This formulation of the density estimation also gives rise to a recursive algorithm for the evaluation of the gradient. The gradient expression to be used in the update at time instant k, in terms of the gradient from the previous samples and the kernel function, now becomes

$$\nabla_\theta^k p = \alpha \nabla_\theta^{k-1} p + (1-\alpha)\nabla\kappa_\sigma(\theta - \theta_k),$$

and is evaluated at the current estimate of the angle θ. In the update phase, only one of the three angles is updated, and that is determined by comparing the difference between the angle of the current sample and the estimates of the angles from the previous update. The distance is measured in mod-π arithmetic since the directions of the columns have a periodicity of π radians.

A number of simulations have been carried on to evaluate the performance of this tracking algorithm. It has been determined that the tracking ability of the algorithm is limited by the first and second derivatives of the angles of the columns with respect to time. Using the values 0.9 for the forgetting factor, $3 \cdot 10^{-3}$ rad for the size of the Gaussian kernel (this is approximately the value obtained for the optimal kernel size as given in figure 2), and a step size of 10^{-7}, the algorithm was able to track signals with first order derivatives on the order of 10^{-5} rad/sample. Figure 4 presents an example of such a simulation. In this example, the directions of the three columns of the mixing matrix are varying in time as sinusoids of various amplitudes and frequencies, adjusted such that their maximum time derivatives do not exceed the determined upper limit. The initial estimates were computed using the batch training algorithm on one hundred samples also collected from the same time-varying mixing matrix.

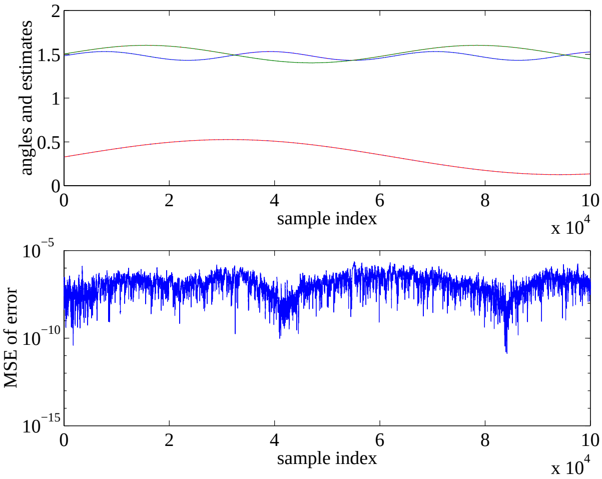

Fig. 4. Tracking of the angles that define the columns of the mixing matrix.

The question at this point is, how good this tracking algorithm in tracking changing mixing matrices in a typical environment is. For example, will it be able to track the changing mixing matrix when a speaker walks around in the room? In order to investigate the answer to these questions, we use a simple spherical sound wave propagation model. Assume that the source signal generated by a point source propagates radially at all directions and the power (intensity) of the signal decreases proportional to the distance squared. In that case the instantaneous power received at sensor i due to the source signal j is written as

$$x_i^2(t) = \frac{\beta^2}{d_{ij}^2} s_j^2(t),$$

where s_j is the jth source signal amplitude, d_{ij} is the distance from sensor i to source j, and β is a coefficient depending on all other environmental factors. Then, in order to be consistent with this power attenuation model, the amplitude of the received signal at sensor i has to be

$$x_i(t) = \frac{\beta}{d_{ij}} s_j(t)$$

Thus the entry a_{ij} of the mixing matrix can be determined from this equation to be

$$a_{ij} = \frac{\beta}{d_{ij}}. \qquad (3)$$

The distance-squared from source j to sensor i is

$$d_{ij}^2 = (\mathbf{s}_j - \mathbf{x}_i)^T(\mathbf{s}_j - \mathbf{x}_i), \qquad (4)$$

and the velocity of source j is

$$\dot{\mathbf{s}}_j = \mathbf{v}_j. \qquad (5)$$

Combining (5) and (4) we get

$$\dot{d}_{ij} = (\mathbf{s}_j - \mathbf{x}_i)^T \frac{\mathbf{v}_j}{d_{ij}}.$$

Combining this with (3) we obtain

$$\dot{a}_{ij} = -\beta(\mathbf{s}_j - \mathbf{x}_i)^T \frac{\mathbf{v}_j}{d_{ij}^3}. \qquad (6)$$

Since the angle of the jth column of the mixing matrix is written as

$$\theta_j = \arctan \frac{a_{2j}}{a_{1j}},$$

taking the time derivative we have

$$\dot{\theta}_j = \frac{\dot{a}_{2j}a_{1j} - \dot{a}_{1j}a_{2j}}{a_{1j}^2 + a_{2j}^2};$$

and finally combining this with (6) and (3), the following relation between the change of directions of the columns of $\mathbf{A}$ and the physical environment is obtained

$$\dot{\theta}_j = \frac{d_{1j}d_{2j}}{d_{1j}^2 + d_{2j}^2}\left[\frac{\mathbf{s}_j - \mathbf{x}_1}{d_{1j}^2} - \frac{\mathbf{s}_j - \mathbf{x}_2}{d_{2j}^2}\right]^T \mathbf{v}_j. \qquad (7)$$

The upper bound for the absolute value of (7) corresponds to the case where the source is aligned with the measurement points and the velocity vector is collinear with them

$$|\dot{\theta}_j| \leq \frac{2d_x}{4d_j^2 + d_x^2}||\mathbf{v}_j|| \equiv U_{\max}(d_x, d_j)||\mathbf{v}_j||,$$

where d_x is the distance between sensors, and d_j is the distance from the midpoint between sensors to source j. Therefore, the worst case upper bound for the velocity of the sources is given by

$$||\mathbf{v}_j|| \leq \frac{|\dot{\theta}_{\max}|f_s}{U_{\max}(d_x, d_j)} \quad \frac{\text{rad/sample} \cdot \text{samples/sec}}{\text{rad/m}}.$$

We know that, with the chosen parameters, the algorithm can track time variations of 10^{-5} rad/sample. Suppose a sampling frequency of 10 KHz is used. Then figure 5 shows the worst case upper bound on the speed of the speaker as a function of distance to microphones for different values of microphone separation.

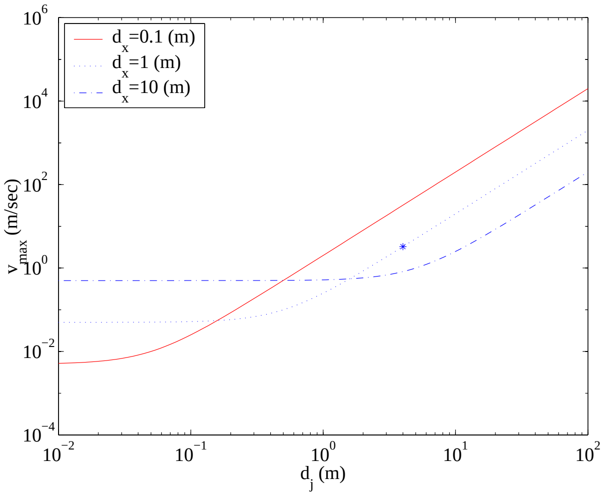

Fig. 5. Worst case upper bound on the velocity of the speaker as a function of distance to microphones for different values of separation between microphones.

For instance, if the speaker is four meters away from the microphones, and the microphones are placed one meter apart, according to the above formula, the worst case upper bound for his speed would be 3.25 m/sec, as is illustrated in figure 5 with a star.

Even though, in our analyses we have considered the underdetermined case with two-measurements, for the sake of simplicity, the presented methodology and algorithms can be easily generalized to cases with higher dimensionality and to squared and overdetermined cases.

5. CONCLUSIONS

In this communication, we have studied the problem of estimating the mixing matrix in the context of instantaneous blind source separation. The approach followed involved determining the peaks of the conditional probability density of the measurement directions given the samples. It has been shown that, even for low sparsity factors, these peaks correspond well to the true directions of the columns of the mixing matrix. The conditional distribution is estimated using Parzen windowing, and the peaks of the histogram are used as initial estimates for determining the peaks of this estimated conditional distribution. This approach is shown to greatly refine the estimation of the columns of the mixing matrix.

Next, the algorithm had been modified to deal with the tracking of the columns in the dynamic case where the mixing matrix is assumed to be time varying. An upper bound on the tracking ability of the algorithm has been determined, and using a simplified wave propagation model, this value had been used to estimate the maximum allowable speed for mobile sources as a function of sensor separation and distance to sources.

Acknowledgments: This work was partially supported by the NSF grant ECS-9900394.

6. REFERENCES

[1] M. Zibulevsky, B. Pearlmutter, P. Bofill, and P. Kisilev, *Independent Components Analysis: Principles and Practice*, ch. Blind source separation by sparse decomposition in a signal dictionary. Cambridge University Press, 2000. In press.

[2] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?,” *Vision Research*, no. 37, pp. 33311–3325, 1997.

[3] J. K. Lin, D. G. Grier, and J. D. Cowan, “Faithful representation of separable distributions,” *Neural Computation*, vol. 9, pp. 1303–1318, 1997.

[4] P. Bofill and M. Zibulevsky, “Underdetermined blind source separation using sparse representations,” *submitted to Signal Processing*, 2000.

[5] H.-C. Wu, *Blind Source Separation using Information Measures in the Time and Frequency Domains*. PhD thesis, CNEL, University of Florida, 1999.

[6] E. Parzen, *Time Series Analysis Papers*, ch. On Estimation of a Probability Density Function and Mode. Holden-Day, 1967.

SOME PROPERTIES OF BELL–SEJNOWSKI PDF-MATCHING NEURON

Simone Fiori

DIE–UNIPG, University of Perugia, Italy
E-MAIL: SFR@UNIPG.IT

ABSTRACT

The aim of the present paper is to investigate the behavior of a single-input single-unit system, learning through the maximum-entropy principle, in order to understand some formal property of Bell-Sejnowski's PDF-matching neuron. The general learning equations are presented and two case-study are discussed with details.

1. INTRODUCTION

The analysis of the behavior of adaptive activation function non-linear neurons is a challenging research field in the neural network theory, which may require analyzing non-linear differential equations of neuron's parameters. Especially in signal processing applications, the external excitations are not deterministic but stochastic, and the aim is to find a statistical description of the neural system's response and of system features. The formal techniques known in the scientific literature for studying such systems benefit from cross-fertilization among artificial neural networks, information theory and signal processing and neurobiology.

Recently, several researchers have focused their attention on this class of stochastic learning theories, with applications to blind separation of sources by the independent component analysis [1, 2, 3, 4, 5, 6, 16, 20], probability density estimation [1, 18, 7], self-organizing classification [19], and blind system deconvolution [2, 8, 9]. Also, some studies on neurobiological mechanisms have suggested interesting non-linear models and information-theoretic based learning theories [10, 11, 13, 14, 15].

Following the pioneering work of Linsker, Plumbley, Bell and Sejnowski [2, 12, 17], in recent papers, we presented some results related to the use of flexible non-linear units, termed FANs, trained in an stochastic way by means of an entropy-based criterion: In [7] we proposed some general structures and adapting frameworks for FAN non-linear unit, while papers [4, 5, 6] have been devoted to the application of these neurons to blind signal processing tasks, such as blind source separation by the independent component analysis and blind signal flattening; in these works we also compared the proposed structures to other flexible topologies known in the scientific literature, as e.g. the mixture-of-kernel, showing that the new approach may exhibit better estimation/approximation ability at a lower complexity burden.

The aim of our preceding work was to introduce the new adaptive-activation-function structures and adapting theories and to assess their features through numerical experiments on real-world data; however, due to the strong non-linearity of the involved equations we did not present any theoretical considerations about the mathematical structure and properties of the adapting equations. In the present paper we recall the basic adapting formulas and present the closed-form expressions of them for some special cases; our main goal is to discuss their features in an analytical way, in order to gain a deeper insight into the behavior of the non-linear differential equations governing information-theoretic FAN non-linear unit adapting, and to better explain the previous numerical results. In particular, the aim is to discuss some properties of Bell-Sejnowski probability density function matching neuron.

2. NEURON MODEL AND PDF-MATCHING LEARNING EQUATIONS

In the present paper we consider the simple neuron model depicted in the Figure 1, which may be formally described by the input-output equation:

$$y = s(wx + b) \ , \tag{1}$$

where $x(t) \in \mathcal{R}$ and $y(t) \in \mathcal{R}$ denote the neuron's input stimulus and the neuron's response signal, respectively, $w \in \mathcal{R}$ denotes the neuron's connection strength and $b \in \mathcal{R}$ stands for the bias; the non-linear function $s(\cdot)$ represents a bounded (saturating) squashing activation function, which meets the monotonicity condition $s'(\cdot) > 0$.

Both the input and output signals are treated as stationary stochastic signals, described by the probability density functions (pdfs) $p_x(x)$ and $p_y(y)$. We do not make any particular hypothesis about the stimulus' statistical distribution, but for requiring a sufficient regularity, namely $p_x(x)$ should be a smooth function endowed with sufficiently-high-order moments.