Flexible estimation of joint probability and joint cumulative density functions

M. Scarpiniti, R. Parisi and A. Uncini

The aim of this reported work is to extend a recent, simple and effective algorithm for the estimation of the probability density function and cumulative density function to the case of bidimensional random vectors. The algorithm is based on an information maximisation approach. The nonlinear bidimensional function involved in the algorithm is adaptively modified during learning and is implemented by using a bidimensional spline function.

Introduction: Joint probability density function (PDF) estimation is a very important issue in several interesting areas, such as blind signal processing and adaptive data processing, and appealing for the ever increasing use of multisensory signals [1]. In this sense the use of an easy and fast method of estimation of the joint PDF or the joint cumulative density function (CDF) becomes a very important task.

Recently a novel, fast and efficient method to estimate CDF and PDF was presented in [2]. The aim of this Letter is to extend the algorithm described in [2] to the case of bidimensional random vectors. We propose the use of a flexible single-input neuron, i.e. a nonlinear bidimensional activation function, the shape of which can be changed during the learning process following the method shown in [3]. This nonlinear function is implemented by a cubic bidimensional spline function. Spline functions consist of a superposition of a certain number of cubic polynomial pieces, so their shape can be varied during the learning process. Several experimental results demonstrate the effectiveness of the proposed approach.

Information maximisation approach: Let $\mathbf{x}(t) = [x_1(t),\ x_2(t)]$ be a stationary bidimensional random process with unknown joint probability density function $p_X(\mathbf{x})$ and let $y = f(\mathbf{x}) = f(x_1, x_2)$, where $f(\ldots)$ is a monotone increasing continuous and bidimensional function. As in [2], the proposed algorithm addresses the problem of maximising the mutual information between the random vector $\mathbf{x}$ and the invertible nonlinear transform y, which is equivalent to maximising the output differential entropy $H(y)$ [4, 5]. In addition, the proposed bidimensional nonlinear function $f(x_1, x_2)$ is implemented in a flexible manner by a bidimensional spline function in order to change the shape of the nonlinearity during the learning process. When the algorithm converges, the shape of the nonlinear function matches the joint CDF of the random vector $\mathbf{x}$. Estimation of the joint CDF is achieved through the system shown in Fig. 1.

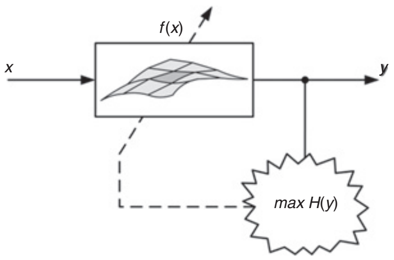

Fig. 1 *System model for joint CDF estimation*

The differential entropy of the system output y is simply obtained by exploiting the relationship between the input and output PDF of a nonlinear transformation $p_Y(y) = p_x(\mathbf{x})/\ \mathbf{J}$, where $\mathbf{J}$ is the Jacobian of the transformation [5]. To render the Jacobian a square matrix, we can add a second variable $\mathbf{y} = [y, z]$, where

$$y = f(x_1, x_2)$$

$$z = \frac{1}{2}(x_2 - x_1)$$

So the Jacobian matrix becomes

$$|\mathbf{J}| = \det\begin{bmatrix} \frac{\partial y}{\partial x_1} & \frac{\partial y}{\partial x_2} \\ \frac{\partial z}{\partial x_1} & \frac{\partial z}{\partial x_2} \end{bmatrix} = \frac{\partial y}{\partial x_1}\frac{\partial z}{\partial x_2} - \frac{\partial y}{\partial x_2}\frac{\partial z}{\partial x_1} = \frac{1}{2}\left(\frac{\partial y}{\partial x_1} + \frac{\partial y}{\partial x_2}\right) \tag{1}$$

In this way, the entropy of the network output y is

$$H(y) = -E\{\mathrm{In}\, p_y(y)\} = H(\mathbf{x}) + E\{\mathrm{In}|\mathbf{J}|\} = H(\mathbf{x}) + \frac{1}{2}E\left\{\mathrm{In}\left|\frac{\partial y}{\partial x_1} + \frac{\partial y}{\partial x_2}\right|\right\} \tag{2}$$

Equation (2) can be interpreted as the Kullback-Leibler divergence between the true density of $\mathbf{x}$, $p_{\mathbf{x}}(\mathbf{x})$ and an arbitrary density given by $\frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2|$:

$$-H(y) = E\left\{\log\frac{2p_{\mathbf{x}}(\mathbf{x})}{|\partial y/\partial x_1 + \partial y/\partial x_2|}\right\} = D\left[p_{\mathbf{x}}(\mathbf{x}), \frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2|\right] \tag{3}$$

It follows that the divergence between $p_{\mathbf{x}}(\mathbf{x})$ and $\frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2|$ is minimised when the entropy $H(y)$ is maximised. Maximisation of (2) is achieved if $\frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2| = p_x(x)$, that is $f(x_1,x_2)$ can be interpreted as the joint CDF of the input source $\mathbf{x}$. In this sense, this system is able to provide an estimate of the joint CDF, hence an estimate of the joint PDF can be obtained as the derivative of the estimated CDF. The main issue is to ensure that $\frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2|$ is a density for $\mathbf{x}$.

We show the following lemma, which is an extension of the one-dimensional case [6].

Lemma 1: Suppose that $y = f(x_1,x_2)$ is a monotone increasing and differentiable function satisfying $\lim_{\nu\to-\infty} f(\nu, \xi) = 0$, $\lim_{\xi\to-\infty} f(\nu, \xi) = 0$, and $\lim_{\substack{\nu\to+\infty \\ \xi\to+\infty}} f(\nu, \xi) = 1$. Then $\frac{1}{2}|\partial y/\partial x_1 + \partial y/\partial x_2|$ is a density of $\mathbf{x}$.

Proof: We have to show that

$$\frac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |\partial y/\partial x_1 + \partial y/\partial x_2| dx_1 dx_2 = 1 \tag{4}$$

Clearly $|\partial y/\partial x_1 + \partial y/\partial x_2| = \partial y/\partial x_1 + \partial y/\partial x_2$ because $f(x_1,x_2)$ is monotone increasing, therefore we obtain

$$\frac{1}{2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\frac{\partial y}{\partial x_1} + \frac{\partial y}{\partial x_2}\right)dx_1 dx_2 = \frac{1}{2}(f(x_1, \xi)|_{-\infty}^{+\infty} + f(\nu, x_2)|_{-\infty}^{+\infty}) = \frac{1}{2} + \frac{1}{2} = 1$$

which completes the proof.

Spline function: The implementation of the flexible function $f(x_1, x_2)$ is reached by a spline interpolation scheme [3]. Splines are smooth parametric curves defined by interpolation of properly defined control points collected in a look-up table. Let $y = f(x_1,x_2)$ be a function to be estimated. In the general case, given N^2 control points equispaced on a regular grid of dimension N, the spline surface results as a concatenation of local $(N-3)^2$ adjacent surfaces. The spline estimation provides an approximation $f(x_1, x_2) \simeq \tilde{y} = \hat{f}(u_1, u_2, i_1, i_2)$ based on two pairs of parameters (u_1,i_1) and (u_2,i_2) directly depending on x_1 and x_2, that can be estimated by equation (4) in [2]. In this specific application, for each input occurrence $(\bar{x}_1, \bar{x}_2)$ the spline estimates $f(\bar{x}_1, \bar{x}_2)$ by using 16 control points selected inside the look-up table [3]. Two points are the adjacent control points on the left side of each $\bar{x}_1$ and $\bar{x}_2$, while two other points are the control points on the right side. Hence the output of a generic input $(\bar{x}_1, \bar{x}_2)$ is simply obtained by the following matrix expression [3]:

$$\bar{y} = f(\bar{x}_1, \bar{x}_2) = \mathbf{T}_2\mathbf{M}(\mathbf{T}_1\mathbf{M}\mathbf{Q}_{i_1,i_2})^T \tag{5}$$

where $\mathbf{T}_k = [u_k^3,\ u_k^2,\ u_k,\ 1]$, $k = 1, 2$, $\mathbf{Q}_{i_1,i_2}$ is the matrix that collects the 16 local control points $q_{i,k}$ and $\mathbf{M}$ is a 4×4 matrix which selects which spline base is used, typically B-spline or Catmull-Rom spline (CR-spline) [3]. To ensure the monotonously increasing characteristic of the overall function, the additional constraint $q_{i,k} < q_{i+1,k}$ must be imposed.

Algorithm derivation: The learning algorithm is derived by maximising (2) and using the expression (5) for the function $f(x_1,x_2)$. The learning rule is local and involves the adaptation of only 16 control points:

$$\begin{aligned}\Delta Q_{i_1+m_1,i_2+m_2} &\propto \frac{\partial H(y)}{\partial Q_{i_1+m_1,i_2+m_2}} = \frac{\partial H(x)}{\partial Q_{i_1+m_1,i_2+m_2}} \\ &\quad + \frac{\partial}{\partial Q_{i_1+m_1,i_2+m_2}} E\left\{\frac{1}{2}\ln\left|\frac{\partial y}{\partial x_1}+\frac{\partial y}{\partial x_2}\right|\right\} \\ &\simeq \frac{\partial}{\partial Q_{i_1+m_1,i_2+m_2}}\left\{\frac{1}{2}\ln\left|\frac{\partial y}{\partial x_1}+\frac{\partial y}{\partial x_2}\right|\right\} \\ &= \frac{1}{2}\frac{1}{\frac{\partial y}{\partial x_1}+\frac{\partial y}{\partial x_2}}\frac{\partial\left\{\frac{\partial y}{\partial x_1}+\frac{\partial y}{\partial x_2}\right\}}{\partial Q_{i_1+m_1,i_2+m_2}} \\ &= \frac{1}{2}\frac{\mathbf{T}_2\mathbf{M}_{m_2}(\dot{\mathbf{T}}_1\mathbf{M}_{m_1})^T + \dot{\mathbf{T}}_2\mathbf{M}_{m_2}(\mathbf{T}_1\mathbf{M}_{m_1})^T}{\mathbf{T}_2\mathbf{M}(\dot{\mathbf{T}}_1\mathbf{MQ})^T + \dot{\mathbf{T}}_2\mathbf{M}(\mathbf{T}_1\mathbf{MQ})^T}; \\ &\quad m_1, m_2 = 0, \ldots, 3\end{aligned} \tag{6}$$

where $\mathbf{M}_m$ is a matrix in which all the elements are zero, except the mth column which is equal to the mth column of the $\mathbf{M}$ matrix and $\dot{\mathbf{T}}_k = [3u_k^2, 2u_k, 1, 0]$, $k = 1, 2$. The relations in (6) are obtained using the approximation of the expectation operator $E\{g(x)\} \simeq g(x)$. The final learning rule for the adaptation of the spline control points using the stochastic gradient algorithm, at instant $l + 1$, is simply

$$Q_{i_1+m_1,i_2+m_2}(l+1) = Q_{i_1+m_1,i_2+m_2}(l) + \eta\Delta Q_{i_1+m_1,i_2+m_2}(l) \tag{7}$$

where η is the learning rate, a small and positive constant.

Results: The proposed approach was tested in blind estimation of the joint CDF and joint PDF of a random vector **x**. Some experimental tests are shown. In the first test a random vector of 10000 samples with a jointly Gaussian distribution with zero mean and unitary variance is chosen. This vector is generated using the *random* function in MATLAB®. We adopt a CR-spline with $N = 43$ control points and a learning rate $\eta = 10^{-7}$. The estimated joint CDF and joint PDF are shown in Fig. 2, which clearly shows the effectiveness of the learning. In the second test a 16 PSK signal of 10000 samples is chosen. The parameters are the same as in the previous example. This test shows the ability of the proposed architecture to estimate the joint PDF, the 16 peaks, in Fig. 3. The Figure confirms the effectiveness of the learning.

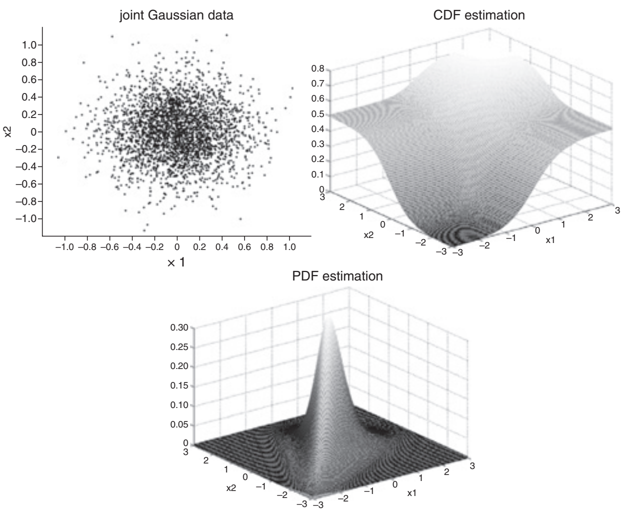

Fig. 2 *Data scatterplot (top left), CDF estimation (top right) and PDF estimation (bottom) of joint Gaussian random vector*

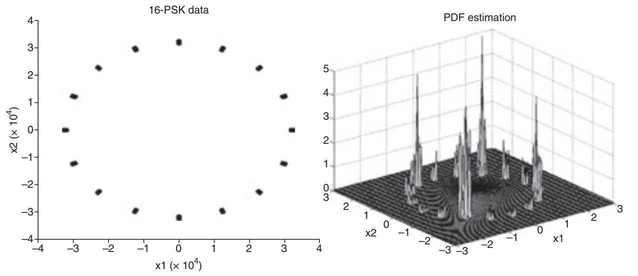

Fig. 3 *Data scatterplot (left) and estimated PDF (right) of 16-PSK source signal*

Conclusion: A novel, fast and efficient method to estimate joint cumulative density function and joint probability density function is presented. The proposed approach is based on the bidimensional spline function: the shape of this surface is adaptively changed during the learning process by maximising the information of a transformed version of the input signal and finally reproducing the profile of the desired functions. Some experimental results demonstrate the effectiveness of the proposed approach.

24 May 2010
doi: 10.1049/el.2010.1307

M. Scarpiniti, R. Parisi and A. Uncini (*INFOCOM Department, 'Sapienza', University of Rome, via Eudossiana 18, Roma 00184, Italy*)
E-mail: michele.scarpiniti@uniroma1.it

References

1 Pan, H., Liang, Z.-P., and Huang, T.S.: 'Estimation of the joint probability of multisensory signals', *Pattern Recognit. Lett.*, 2001, **22**, (13), pp. 1431–1437
2 Scarpiniti, M., Parisi, R., and Uncini, A.: 'Flexible estimation of probability and cumulative density functions', *Electron. Lett.*, 2009, **45**, (21), pp. 1095–1096
3 Solazzi, M., and Uncini, A.: 'Regularising neural networks using flexible multivariate activation function', *Neural Netw.*, 2004, **17**, (2), pp. 247–260
4 Bell, A.J., and Sejnowski, T.J.: 'An information-maximisation approach to blind separation and blind deconvolution', *Neural Comput.*, 1995, **7**, pp. 1129–1159
5 Papoulis, A.: 'Probability, random variables and stochastic processes' (McGraw-Hill, 1991)
6 Roth, Z., and Baram, Y.: 'Multidimensional density shaping by sigmoids', *IEEE Trans. Neural Netw.*, 1996, **7**, (5), pp. 1291–1298