Flexible estimation of probability and cumulative density functions

M. Scarpiniti, R. Parisi and A. Uncini

A novel, simple and effective algorithm for the estimation of the probability density function and cumulative density function is presented. The algorithm is based on an information maximisation approach. The nonlinear function involved in the algorithm is adaptively modified during learning and is implemented by using a spline function.

Introduction: Probability density function (PDF) estimation is a very important issue in several interesting areas, such as blind signal processing and adaptive data processing. The estimation of the PDF or the cumulative density function (CDF) through use of an easy and fast method becomes a very important task.

Several approaches exist [1] such as maximum likelihood estimation, kernel estimation, cluster analysis and the most important histogram method. An effective method is based on neural networks and provides a simple adaptation rule. An example can be found in [2] that is based on the information-theoretic approach proposed by Bell and Sejnowski [3].

In this Letter we propose the use of a single-input neuron, i.e. the nonlinear activation function, by adopting a flexible nonlinear function the shape of which can be changed during the learning process following the method shown by Solazzi *et al.* in [4]. This nonlinear function is implemented by a cubic 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. Some experimental results are shown in order to demonstrate the effectiveness of the proposed approach.

Information maximisation approach: Let x(t) be a stationary random process with unknown probability density function pX(x) and let y = f(x), where f(.) is a monotone increasing continuous function. The main idea of the proposed approach is based on the well-known Infomax principle [3]. This algorithm addresses the problem of maximising the mutual information between the random vector x and the invertible nonlinear transform of it, y. Because this is a deterministic mapping, maximising the mutual information is the same thing as maximising the output differential entropy H(y) [3, 5]. In addition, the proposed nonlinear function f(x) is implemented in a flexible manner by a spline function. This is because we are interested in changing the shape of the nonlinearity during the learning process. When the algorithm converges, the shape of the nonlinear function is matching the CDF of the random vector x, as shown below. Estimation of the CDF is achieved through the system shown in Fig. 1.

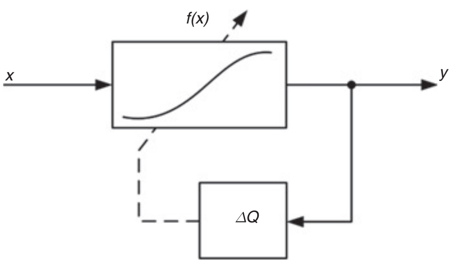

Fig. 1 *System model for CDF estimation*

The differential entropy of the system output y is simply obtained by exploiting the relationship [5] between the input and output PDF of a nonlinear transformation $p_Y(y) = p_X(x)/|f'(x)|$:

$$H(y) = -E\{\log p_Y(y)\} = -E\left\{\log \frac{p_X(x)}{|f'(x)|}\right\} = H(x) + E\{\log f'(x)\} \quad (1)$$

Equation (1) can be interpreted as the Kullback-Leibler divergence between the true density of x, $p_X(x)$ and an arbitrary density given by $|f(x)|$:

$$-H(y) = E\left\{\log \frac{p_X(x)}{|f'(x)|}\right\} = D[p_X(x), |f'(x)|] \quad (2)$$

It follows that the divergence between $p_X(x)$ and $|f'(x)|$ is minimised when the entropy H(y) is maximised. The maximisation of (1) is achieved if $|f'(x)| = p_x(x)$, i.e. f(x) is matching the CDF of the input source x. In this sense, if one uses a flexible function f(x) as activation function, this system is able to provide an estimate of the CDF. Hence an estimate of the PDF can be obtained as the derivative of the estimated CDF.

The main issue is to assure that $|f'(x)|$ is a density for x. Let us show the following [6]:

Lemma 1: Suppose that f(.) is a monotone increasing and differentiable function satisfying $\lim_{v\to\infty} f(v) = 0$ and $\lim_{v\to\infty} f(v) = 1$ and let y = f(x). Then $|f'(x)|$ is a density of x.

Proof: We have to show that

$$\int_{-\infty}^{\infty} |f'(x)| dx = 1 \quad (3)$$

Clearly $|f'(x)| = f'(x)$ because f(x) is monotone increasing (f′(x) > 0), therefore we obtain

$$\int_{-\infty}^{\infty} |f'(x)| dx = f(v)|_{-\infty}^{\infty} = 1$$

which demonstrates (3) and completes the proof.

Spline function: The implementation of the flexible function *f*(*x*) is reached by a spline interpolation scheme [4]. Splines are smooth parametric curves defined by interpolation of properly defined control points collected in a look-up table. In the general case, given N equispaced control points, the spline curve results as a polynomial interpolation through N − 3 adjacent spans. Let y = *f*(*x*) be some function to be estimated. The spline estimation provides an approximation $f(x) \cong \tilde{y} = \hat{f}(u(x), i(x))$ based on two parameters (u,i) directly depending on x:

$$\begin{aligned} z' &= \frac{x}{\Delta} + \frac{N-2}{2} \\ z &= \begin{cases} 1 & z' \\ z' & 1 \le z' \le N-3 \\ N-3 & z' > N-3 \end{cases} \\ i &= \lfloor z \rfloor \\ u &= z - i \end{aligned} \quad (4)$$

where z, z′ are two internal variables and Δx is the distance between two consecutive control points, qi and q_{i+1}. In this specific application, for each input occurrence $\bar{x}$ the spline estimates $f(\bar{x})$ by using four control points selected inside the look-up table [4]. Two points are the adjacent control points on the left side of $\bar{x}$, while the other two points are the two control points on the right side.

Hence the output of a generic input $\bar{x}$ is simply obtained by the following matrix expression, as explained in detail in [4]:

$$\bar{y} = f(\bar{x}) = \hat{f}(u(\bar{x}), i(\bar{x})) = \mathbf{TMQ}_i \quad (5)$$

where $\mathbf{T} = [u^3\, u^2\, u 1]$, $\mathbf{Q} = [q_i\, q_{i+1}\, q_{i+2}\, q_{i+3}]^T$ is the vector that collects the local control points and **M** is a 4×4 matrix which selects which spline base is used, typically B-spline or Catmull-Rom spline (CR-Spline) [4].

To ensure the monotonously increasing characteristic of the overall function, the additional constraint $q_i < q_{i+1}$ must be imposed.

Algorithm derivation: The learning algorithm is derived by maximising (1) using expression (5) for the function *f*(*x*). The learning rule is local and involves the adaptation of only four control points for each input sample:

$$\begin{aligned} \Delta Q_{i+m} &\propto \frac{\partial H(y)}{\partial Q_{i+m}} = \frac{\partial H(x)}{\partial Q_{i+m}} + \frac{\partial}{\partial Q_{i+m}} E\{\log f'(x)\} = \\ &\cong \frac{\partial}{\partial Q_{i+m}} \log f'(x) = \frac{1}{f'(x)} \frac{\partial f'(x)}{\partial Q_{i+m}} \\ &= \frac{\Delta x}{\dot{\mathbf{T}}\mathbf{MQ}_i} \frac{\partial(\dot{\mathbf{T}}\mathbf{MQ}_i)}{\Delta x \partial Q_{i+m}} = \frac{\dot{\mathbf{T}}\mathbf{M}_m}{\dot{\mathbf{T}}\mathbf{MQ}_i}; \quad m = 0, \ldots, 3 \end{aligned} \quad (6)$$

where $\mathbf{M}_m$ is the mth column of the $\mathbf{M}$ matrix, $\dot{\mathbf{T}} = [3u^2\ 2u\ 1\ 0]$. The relations in (6) are obtained by using the approximation of the expectation operator $E\{\log f'(x)\} \simeq \log f'(x)$. The final learning rule for the adaptation of the spline control points using the stochastic gradient algorithm at sample k + 1, is simply

$$Q_{i+m}(k+1) = Q_{i+m}(k) + \eta \frac{\dot{\mathbf{T}}\mathbf{M}_m}{\dot{\mathbf{T}}\mathbf{M}\mathbf{Q}_i} \quad (7)$$

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

Results: The proposed approach was tested in blind estimation of the CDF and PDF of a random source x. Some experimental tests are shown. In the first test a random vector of 10000 samples with a uniform distribution in [0 1] is chosen. This vector is generated using the *rand* function in MATLAB®. We adopt a CR-spline with N = 43 control points and a learning rate $\eta = 10^{-4}$, selected heuristically in order to obtain the best performances. The estimated CDF and PDF are shown in Fig. 2, which clearly shows the effectiveness of the learning. In a second test a random vector of 10000 samples with an exponential distribution is chosen. This vector is generated using the *random* function in MATLAB®. The estimated CDF and PDF, using the same parameters of the previous example, but adopting a B-spline, are shown in Fig. 3, which clearly shows the effectiveness of the learning.

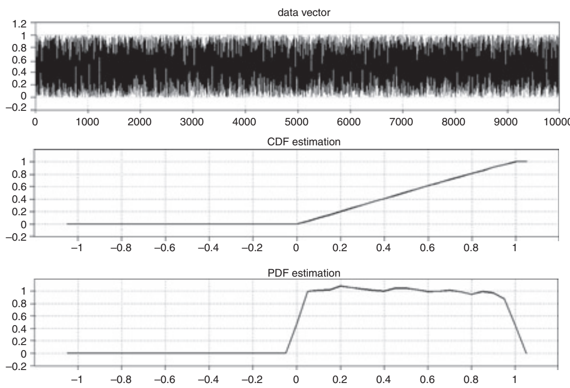

Fig. 2 *Data vector (first row), CDF (second row), PDF (third row) estimation of uniform random variable*

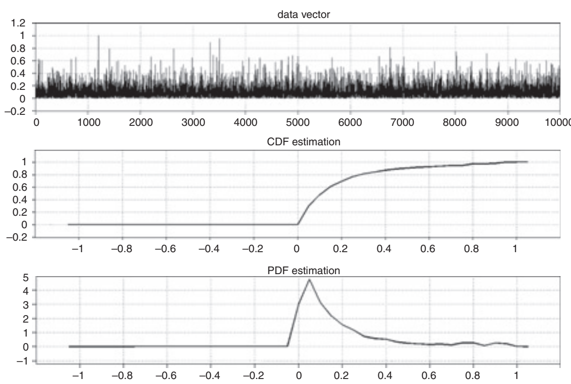

Fig. 3 *Data vector (first row), CDF (second row), PDF (third row) estimation of exponential random variable*

Conclusion: A novel, fast and efficient method to estimate cumulative density function and probability density function is presented. The proposed approach is based on the use of a spline function as activation function: the shape of this function is adaptively changed during the learning process by maximising the entropy of a nonlinearly transformed signal and finally reproducing the profile of the desired CDF. Some experimental results demonstrate the effectiveness of the proposed approach.

5 May 2009
doi: 10.1049/el.2009.1274

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 Silverman, B.W.: 'Density estimation for statistical data analysis' (Chapman & Hall, 1986)
2 Fiori, S., and Bucciarelli, P.: 'Probability density estimation using adaptive activation function neurons', *Neural Process. Lett.*, 2001, **13**, (1), pp. 31–42
3 Bell, A.J., and Sejnowski, T.J.: 'An information-maximisation approach to blind separation and blind deconvolution', *Neural Comput.*, 1995, **7**, pp. 1129–1159
4 Solazzi, M., Piazza, F., and Uncini, A.: 'An adaptive spline non-linear function for blind signal processing'. Proc. of Neural Networks for Signal Processing, 2000, Vol. 1, pp. 396–404
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