EURASIP Journal on Applied Signal Processing 2001:2, 121–127

Robust Control Methods for On-Line Statistical Learning

Enrico Capobianco

CWI, Kruislaan 413, Amsterdam 1098-SJ, The Netherlands
Email: enrico@cwi.nl

Received 8 January 2001 and in revised form 19 April 2001

The issue of controlling that data processing in an experiment results not affected by the presence of outliers is relevant for statistical control and learning studies. Learning schemes should thus be tested for their capacity of handling outliers in the observed training set so to achieve reliable estimates with respect to the crucial bias and variance aspects. We describe possible ways of endowing neural networks with statistically robust properties by defining feasible error criteria. It is convenient to cast neural nets in state space representations and apply both Kalman filter and stochastic approximation procedures in order to suggest statistically robustified solutions for on-line learning.

Keywords and phrases: artificial learning, statistical control algorithms, robustness and efficiency of estimators, maximum likelihood inference.

1. INTRODUCTION

This work represents, to our knowledge, one of the few attempts done to endow artificial neural networks learning schemes with statistical robustness properties. It is of great interest to realize why the statistical robustness field has not found a wide ground of applications outside statistics, given the fact that more and more applications of algorithms proposed by researchers in different fields are similar in the data sets they use and the goals of the analysis, usually prediction, one example being the very popular time series competitions.

In many cases of empirical work dealing with a modelling experiment, one often finds out from the diagnostic checks run on the observed data that there is enough evidence for not relying on the convenient Gaussian probability distribution which characterize the disturbances driving the stochastic processes and thus their observed realizations. This fact may be due to the presence of outliers, occurring with small probability, or because of a particular nature of the *data generating process* underlying the data, like in financial time series. Simply ignoring the deviations from the Gaussian distribution is one way of conducting inference, which is feasible when these deviations are mild. But a stronger evidence for rejecting normality should bring the researcher to consider more robust statistical learning procedures, suitable to deliver more reliable parameter estimates and model predictions.

In Section 2, we briefly present some well-known algorithms and their relations, starting from the stochastic approximation scheme. In Section 3, the likelihood-based standard inference and related quasi-likelihood ideas are presented, while in Section 4 the above framework is robustified with the definition of M-estimators. In Section 5, some examples of useful M-estimation applications are introduced, and together with the loss functions the correspondent influence functions are given. Section 6 is for the conclusions and the appendix shows how to analytically and efficiently compute first and second derivatives of the likelihood function, whose usage is required in some of the algorithms which were presented.

2. STOCHASTIC APPROXIMATION

In this section, we introduce one of the most famous statistical algorithms. Then, we show its relations with other well-known numerical and learning schemes. Both on-line, that is, recursive, and off-line, that is, batch, procedures are designed and compared. A *state space representation* is then introduced for casting the neural networks learning algorithms into an on-line, through a Kalman filter, parameter estimation scheme. Modified versions of these procedures are given to include relevant features coming from high-order statistics information. If it is true that the systems where our procedure operate are produced with the aim of delivering forecasts for future values of the variables characterizing the system dynamics, it also seems that from a statistical viewpoint and in order to make optimal predictions, model identification and estimation aspects must be considered.

Robust procedures allow us to look at the important aspects mentioned above without being trapped in the parametric constraints which often limit the analyst because they do not reflect the possible departures from the context where phenomena are measured or observed. We start by considering a nonlinear function $f(X_t, \theta)$, where $f : \mathbb{R}^k \times \Theta \to \mathbb{R}$, X_t

is a $k \times 1$ random input vector and $\theta \in \Theta \subset \mathbb{R}^p$ represents the vector of unknown parameters. We may use this setup for forecasting the random variable y_t through a single hidden layer feedforward network structure

$$f(X, \theta) = \alpha_0 + \sum_{j=1}^{q} \alpha_j F(X'\beta_j), \tag{1}$$

where $\theta = (\alpha', \beta')'$ represents the weight vector and $F : \mathbb{R} \to \mathbb{R}$ is a bounded and continuosly differentiable activation function localized at the hidden layers. At the output level we set an identity matrix. One can clearly see that $f(X, \theta)$ is an approximation of the objective function here defined at time t as the conditional expectation of the dependent variable y with respect to the input vector, that is, $g(X_t) = E(y_t/X_t)$. Thus, in this *nonlinear least square* frame we seek a solution θ^* to the problem $\min_\theta[E([y_t - f(X_t, \theta)]^2)]$, or, equivalently, to the *first-order conditions* equation $E(\nabla_\theta f(X_t, \theta)[y_t - f(X_t, \theta)]) = 0$, with ∇_θ representing the gradient $k \times 1$ vector calculated with respect to θ. The Robbins-Monro (RM) *stochastic approximation* (SA) algorithm of [1] stores the current approximation value X_t and approximates the objective function locally and linearly via its gradient. It can also be adapted for a nonlinear regression

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \delta_t \nabla_\theta f(X_t, \hat{\theta}_t)[y_t - f(X_t, \hat{\theta}_t)]. \tag{2}$$

Since this recursion is equivalent to that of a *stochastic gradient* method, it generalizes the well-known *backpropagation* (BP) algorithm [2], popular in neural network learning theory, by allowing for a time varying learning rate.

In batch identification algorithms the prediction error is the building block for the chosen optimization criterion, that is, $\text{Loss}_N(\theta) = (1/N)\sum_{t=1}^{N} \epsilon'_{t\theta}\epsilon_{t\theta}$, where $\epsilon_{t\theta} = y_t - \hat{y}_{t\theta}$. Learning schemes seek to minimize the loss function via iterations according to the negative gradient direction, that is, the steepest descent algorithm, or a Gauss-Newton search direction. Alternatively, we could find the prediction error ϵ_t at each step, in a recursive fashion. In [3], some modifications to the RM algorithm are presented in order to speed up the convergence rate; by inserting a Gauss-Newton step at each updating stage one obtains a *modified Robbins-Monro* algorithm

$$\begin{aligned} \hat{\theta}_{t+1} &= \hat{\theta}_t + \delta_t \hat{L}_{t+1}^{-1} \nabla_\theta f(X_t, \hat{\theta}_t)[y_t - f(X_t, \hat{\theta}_t)], \\ \hat{L}_{t+1} &= \hat{L}_t + \delta_t[\nabla_\theta f(X_t, \hat{\theta}_t)' \nabla_\theta f(X_t, \hat{\theta}_t) - \hat{L}_t], \end{aligned} \tag{3}$$

where now we define $\Xi = ((\text{vec}\, L)', \theta')'$ as the new augmented parameter vector.

The approximation to $E(y_t/X_t)$ is only locally optimal, but it is nevertheless important to relax the usually retained i.i.d. (independently and identically distributed) assumption about the stochastic process generating the data. Thus, both batch and recursive algorithms converge to a local minimum with probability one, that is, $\hat{\theta}_t \to \theta^*$ local minimum of $\lim_t E[y_t - f(x_t, \theta)]^2$; so does the BP algorithm therefore.

Artificial neural networks can be cast in a state space representation and, as in the case of SA algorithms, this is a way to allow for a generalized BP to learn how to approximate nonlinear functions in a time series setup. By considering a k-layered feedforward network structure [4]

$$i_j^k = \sum_{l=1}^{N_{k-1}} \theta_{lj}^{k-1,k} o_l^{k-1} + \beta_j^k, \tag{4}$$

where the input of the jth node in the kth layer is given by the sum of the product of the connection weight θ with the output and a bias parameter β, and the output is a function of the input through $F : \mathbb{R} \to \mathbb{R}$, $o_j^k = F(i_j^k)$, we have an extension of the previous trivial system. From

$$\theta_{lj}^{k-1,k}(t+1) = \theta_{lj}^{k-1,k}(t) - \delta\epsilon_j^k(t) o_l^{k-1}(t), \tag{5}$$

where δ is the learning rate and $\epsilon_j^k = (o_j^k - y_j^k)F^1(i_j^k)$, with F^1 first derivative of the output function, we can rewrite in a more compact fashion the network, that is, in vectorial state space form as

$$\Theta_{t+1} = \Theta_t + G\xi_t, \qquad y_t = O_t(\Theta_t) + \eta_t, \tag{6}$$

where η_t is the output error, $G\xi_t$ is equal to the correction term of the BP recursion (6), with the matrix G which separates in a convenient way the deterministic components from the purely random ones [4], and with only ξ_t characterized by pure erratic behavior. The $O_t(\Theta_t)$ term is written in this way to stress its "state dependence," but it could be reported otherwise as O_{t,Θ_t}.

In order to understand how in fact the Kalman filter algorithm relates to the BP expressed in this reformatted neural net setup, we consider a functional mapping that links input and output layers encompassing the hidden layers, that is, $y_i = h(\theta, x_i) + v_i$, at the generic time i. Here we have the input x_i entering the function h, which also depends on the weight parameters θ that includes the input-to-hidden, hidden-to-hidden and hidden-to-output multi-layers connections; we can represent them in a compact vectorial expression with $\theta = [W'_{(1)}; W'_{(2)}; \dots; W'_{(N)}]$ for $N-1$ hidden layers available; v_i is a disturbance.

A simple linearization of $h(\cdot)$ about θ_i helps to understand the various relations we are trying to enlighten, in the sense that it gives the matrix expression $H_i = \partial h(\theta_i, x_i)/\partial\theta_i$, where $\partial\theta_i = \partial\theta|_{\theta=\theta_i}$; clearly, the step which is required before is $h(\theta, x_i) \approx h(\theta_i, x_i) + H_i^T(\theta - \theta_i)$. Thus, we can rewrite a new linearized observation equation $\dot{y}_i = H_i'\theta + v_i$, which allows for the following *weighted least squares* criterion function to be minimized, $\text{LS}_c = (1/k)\sum_{i=1}^{k} \alpha_i[\dot{y}_i - H_i'\theta]^2$.

The off-line solution of the minimization problem is $\theta_{i+1} = [\sum_{i=1}^{k} \alpha_i H_i H_i']^{-1} \sum_{i=1}^{k} \alpha_i H_i \dot{y}_i$. If the term in the squared parenthesis of this equation is defined as the covariance matrix S_i, a recursive equation can be derived, that is, $S_i = S_{i-1} + \alpha_i H_i H_i^T$. From the off-line estimate θ_{i+1} given above, we can recover a recursive equation expressed by $\theta_{i+1} = S_i^{-1}(S_{i-1}\theta_i + \alpha_i H_i \dot{y}_i)$; by substituting the expressions given before for $\dot{y}_i$ and S_{i-1}, we get the recursion

$$\theta_{i+1} = \theta_i + \alpha_i S_i^{-1} H_i[y_i - h(\theta_i, x_i)]. \tag{7}$$

We can recognize the SA update rule reported in the modified RM algorithm, given a particular choice of the matrix S_i and with $H_i=\nabla_\theta f(\cdot,\cdot)$. Without the S_i matrix in the above *recursive least squares* step, we are left back to the generalized delta rule or BP algorithm. The matrix H_i can be expressed in terms of the weights of the neural net; in particular, given the definition of this matrix, we have the following expression for it:

$$H_i = \left[\left(\frac{\partial \hat{y}_i' |_{\theta_i}}{\partial W_{(1)}'} \right)'; \left(\frac{\partial \hat{y}_i' |_{\theta_i}}{\partial W_{(2)}'} \right)'; \dots ; \left(\frac{\partial \hat{y}_i' |_{\theta_i}}{\partial W_{(N)}'} \right)' \right]'. \tag{8}$$

In this framework we can then apply the *extended Kalman filter* (EKF) [5, 6] and/or the *iterated Kalman filter* (IKF) [7]. By including more terms in the Taylor series expansions we can obtain second-order extended Kalman filters, when $h_t(x_t)$ can be linearized about the updated conditional mean estimate $\hat{x}_{t/t}$. The EKF algorithm is given by

$$\begin{aligned} \hat{x}^+ &= \hat{x}^- + K(y - h(\hat{x}^-)), \qquad P^+ = (I - KH)P, \\ H &= h^1(\hat{x}^-), \qquad K = PH'(HPH' + R)^{-1}, \end{aligned} \tag{9}$$

where P is the a priori value of the covariance matrix of estimation uncertainty, that is, from the system dynamic model, and R stands for the error covariance for the measurement equation. Given $\hat{x} = \hat{x}^-$, we can obtain a more accurate algorithm, the IKF, as follows:

$$\begin{aligned} x_{it+1} &= \hat{x} + K_{it}(y - h(x_{it}) - H_{it}(\hat{x} - x_{it})), \\ P_{it+1} &= (I - K_{it}H_{it})P_{it}, \\ H_{it} &= h^1(x_{it}), \\ K_{it} &= P_{it}H_{it}'(H_{it}P_{it}H_{it}' + R)^{-1}. \end{aligned} \tag{10}$$

The IKF algorithm is indeed an application of the Gauss-Newton (GN) method [7]. It is common in statistics and econometrics to work with estimators that aim at minimizing sums of squared residuals like $S(\theta) = \sum_t \epsilon_t^2$, whose Gradient is $v(\theta) = \partial S(\theta)/\partial\theta = 2\sum(\partial\epsilon_t/\partial\theta)\epsilon_t$ and whose Hessian is $V(\theta) = \partial^2 S(\theta)/\partial\theta\partial\theta' = 2\sum[(\partial\epsilon_t/\partial\theta)(\partial\epsilon_t/\partial\theta') - (\partial^2\epsilon_t/\partial\theta\partial\theta')\epsilon_t]$. Several schemes are able of iteratively finding a solution to the initial minimization problem [8]. The most general one is the *Newton-Raphson* (NR) method, which is given by

$$\theta^* = \hat{\theta} - \left[\sum \left(\frac{\partial\epsilon_t}{\partial\theta}\frac{\partial\epsilon_t}{\partial\theta'} - \frac{\partial^2\epsilon_t}{\partial\theta\partial\theta'}\epsilon_t \right) \right]^{-1} \sum \frac{\partial\epsilon_t}{\partial\theta}\epsilon_t. \tag{11}$$

Since the term involving second derivatives is usually small when compared to the first derivatives product term, the GN scheme approximates the above iterative solution and presents a formula that is identical to NR, apart from the term with second derivatives.

3. LIKELIHOOD-BASED INFERENCE

Once a neural network architecture is cast in a state space representation, the most important aspect is that from the Kalman filter algorithm and its variants we straightforwardly obtain the likelihood function through the *prediction error decomposition* (PED) of [9]. The likelihood function for the whole time series, when temporally dependent observations are considered, is obtained by the joint conditional probability density function, that is, $L(y,\theta) = \Pi_{t=1}^N p(y_t/Y_{t-1})$ (by considering Y_{t-1} the set of observations up to and including y_{t-1}).

Since the innovation or prediction error computed by the filter is $\eta_t = y_t - E(y_t/Y_{t-1})$ and $\text{var}(\eta_t) = D_t$, when the observations are normally distributed the likelihood function can be expressed in terms of the innovations. Therefore, the PED likelihood function, in the general multivariate form, is

$$\log L = -c - \frac{1}{2}\sum_{t=1}^N \log|D_t| - \frac{1}{2}\sum_{t=1}^N \eta_t' D_t^{-1}\eta_t \tag{12}$$

with $c = (KN/2)\log 2\pi$, η_t a $k\times 1$ vector and N observations. Note that under Gaussianity the filter delivers an optimal *minimum mean squared* solution for the estimation problem; under hypotheses different from the Gaussian, the filter gives only a *minimum mean square linear* solution and the values which are computed are *Quasi maximum likelihood* (QML) estimates, less efficient but consistent (and therefore useful to start a recursive or multi-step procedure).

For the case we study here, the solution is therefore suboptimal and close to the optimal one according to the accuracy of the approximation involved. From the likelihood function as given by (12), we can calculate its derivatives either analytically or numerically, and in both cases we can leave the filter to compute these quantities together with the other ones representing the core of the algorithm. For instance, the ith element of the Score vector $\partial\log L/\partial\theta_i$ is given by (see [10] and the appendix for more details)

$$-\frac{1}{2}\sum_t \left[\text{tr}\left[\left(D_t^{-1}\frac{\partial D_t}{\partial\theta_i} \right)(I - D_t^{-1}\eta_t\eta_t') \right] - \frac{\partial\eta_t'}{\partial\theta_i}D_t^{-1}\eta_t \right], \tag{13}$$

therefore, requiring the evaluation of the $k\times k$ matrices of derivatives $\partial D_t/\partial\theta_i$ and the $k\times 1$ vector of derivatives $\partial\eta_t/\partial\theta_i$, for $i = 1,\dots,p$ and $t = 1,\dots,N$. These derivatives may be computed through p additional passes of the Kalman filter; if we consider a new run of the filter with $\theta = [\theta_1\cdots\theta_i + \delta_i\cdots\theta_p]$, we obtain a new set of innovations $\eta_t^{(i)}$ and variances $D_t^{(i)}$ and the numerical approximations of the derivatives are $\delta_i^{-1}[\eta_t^{(i)} - \eta_t]$ and $\delta_i^{-1}[D_t^{(i)} - D_t]$. However, for large complex networks, computations become heavy.

If we denote with $\hat{\theta}$ our maximum likelihood estimate for θ and $\text{IM}(\theta)$ represents the *Fisher information matrix* $\int(\partial\log L/\partial\theta)^2\,dF$ such that $\lim(1/N)\text{IM}(\theta) = I^*(\theta)$, then statistical theory says that under regularity conditions the $\sqrt{N}(\hat{\theta} - \theta) \sim \text{Gaussian}(0,[I^*(\theta)]^{-1})$ law goes through asymptotically. Therefore, $\text{IM}(\theta)$ is a crucial quantity because it gives an estimate of the asymptotic covariance matrix of the maximum likelihood estimator.

In the state space filter setup the PED likelihood function yields an $\text{IM}(\theta)$ which depends on first derivatives only.

With the Kalman filter we can calculate these first derivatives, numerically or analytically, but always through parallel computations with respect to the main algorithm, and then the $\mathrm{IM}(\theta)$ too, quite simply, given the structure of it once it is derived through the prediction errors or innovations (see the appendix).

One problem is to train a network to learn tasks such as approximating an unknown complex function in a different fashion from the one implied by the backpropagation paradigm, which depends from the gradient calculation and therefore requires a precise derivation of it from the loss function. In order to compute the gradient, one can also find an approximation based on the loss function $L(\cdot)$ values calculated at levels represented by "perturbed" parameters, that is, $\hat{\theta}_{k-1} \pm \epsilon_k \Delta_k$ by simply adopting a finite difference method, that is, one based on values like $\hat{L}_k^+ - \hat{L}_k^- / 2\epsilon_k \Delta_{kp}$, for all the parameter vector components simultaneously perturbed.

It basically corresponds to what we have proposed above with the Kalman filter numerical estimates; these estimates are more efficient because they were obtained as a by-product of the filter runs and, moreover, their computation comes from the PED likelihood or quasi-likelihood function and related derivatives, which benefits of the inherited asymptotic properties of such estimates.

When accurate estimates are seeked, then the Berndt, Hall, Hall, and Hausman (BHHH) algorithm, which suggests the approximation of the Hessian by the well-known outer-product formula, that is, $\sum_{t=1}^{N} (\partial \log L_t / \partial \theta)(\partial \log L_t / \partial \theta')$, is the most indicated choice we have in order to improve, at least asymptotically, the efficiency of the initial GN or equivalently IKF estimator $\hat{\theta}$ according to the recursion

$$\theta^* = \hat{\theta} + \lambda \left[\sum_{t=1}^{N} \frac{\partial \log L_t'}{\partial \theta} \frac{\partial \log L_t}{\partial \theta'} \right]^{-1} \sum_{t=1}^{N} \frac{\partial \log L_t}{\partial \theta}, \qquad (14)$$

where λ is a variable step length chosen so that the likelihood function is maximized in a given direction.

4. ROBUST LIKELIHOOD SETUP

Robust methods represent a potential solution to the problem of finding a statistical inference tool that allows for a more flexible model to be built so to represent system dynamics where the variables involved can deviate from fixed assumptions about the probabilistic laws behind. The standard alternative of a fully parametric model has to rely on given probabilistic assumptions, and it is usually convenient to adopt the Gaussian laws to derive the random variables, at the cost of completely misleading inference results in case of deviations from the retained hypotheses.

As an alternative one could adopt a nonparametric setup, where no assumptions are made about the probability laws behind the variables, or a semi-parametric frame, where some parametric assumptions are made for some variables in the systems, but not for all of them, thus leaving some variables to belong in a so-called infinite-dimensional space. In this last case, some efficiency for the estimates is lost compared to the parametric model, when the latter is the correct choice, but the advantage is that consistent solutions are obtained in those circumstances realistically far from being satisfied by parametric assumptions.

We consider the fact that through the likelihood function $L(y, \theta)$ we are able to characterize a taxonomy of models only differentiated by the sample size and the dimension of the parameter space where the optimization must be carried out. A straightforward extension of this idea comes from considering the so-called *M-estimators*, that is, estimators of the "maximum likelihood type," as originally defined in the work of [11, 12], for they depend on an unspecified functional form involved in the objective function to be optimized.

Definition 1 (*M*-estimators). Any estimator θ_n defined by a minimum problem such as $\sum_i \rho(x_i, \theta_n) = \min$ or, equivalently, by the equation $\sum_i \psi(x_i, \theta_n) = 0$, with ρ an arbitrary function and ψ its derivative.

Consider an i.i.d. sample $x_1, \ldots, \ldots x_N$ with distribution function $F \in \mathcal{P}$, where $\mathcal{P}$ is the space of probability measures and θ a statistics or linear functional. We introduce [11] the following.

Definition 2 (influence function (IF)). For $G = \delta_x$, where δ_x is the unit point mass at x, and given $0 \le t \le 1$, we have

$$\mathrm{IF}(x, F, \theta) = \lim_{t \to 0} \frac{\theta[(1-t)F + t\delta_x] - \theta F}{t}. \qquad (15)$$

This formula says that the IF quantifies the effect of an infinitesimal contamination on a statistic θ from one additional observation with value x and the sample size $n \to \infty$ [12]. If the above limit exists, IF is the derivative of the statistic at the underlying distribution function F and, therefore, it is a fundamental quantitative robustness information because apart from assessing the influence of individual data points on the value of an estimate, thus measuring the asymptotic bias, it also gives an explicit formula for the asymptotic variance of the estimate, usually expressed as $\int \mathrm{IF}(x, F \cdot \theta)^2 F(dx)$, thus describing its asymptotic properties. When ρ equals the likelihood function $-\log f(x, \theta)$, we find the same likelihood-based criteria as before.

The IF for *M*-estimators can be calculated, in summary, as follows: one starts from the equation $\int \psi(x, F, \theta) = 0$ and replaces F with a contaminated distribution function, that is, $F_t = (1-t)F + tG$. Then one defines $\dot{\theta} = \lim_{t \to 0}(\theta(F_t) - \theta(F)/t)$ and differentiates the initial equation. After some calculations the following relation is found:

$$\mathrm{IF}^*(x, F, \theta) = \frac{\psi(x, F, \theta)}{-\int (\partial / \partial \theta) \psi(x, F, \theta) F(dx)}. \qquad (16)$$

Therefore, the influence function for *M*-estimators is proportional to the functional ψ, or otherwise the score function, in the likelihood setup. Moreover, one finds an asymptotically efficient estimate only when $\mathrm{IF}(x, F, \theta) = (\partial \log L / \partial \theta)(1/\mathrm{IM}(\theta))$ and it can be shown [12] that the asymptotic variance AV of the estimate is such that AV =

$\int \mathrm{IF}(x,F,\theta)^2\,dF \geq 1/\mathrm{IM}(\theta)$, the equality holding only for $\mathrm{IF} \propto (\partial \log L/\partial\theta)$.

Definition 3 (one-step *M*-estimators). Estimators that solve empirical moment equations like $\sum_i \psi(x_i,\theta_n)/n = 0$; for $\Psi = \partial E(\psi(x,\theta)/\partial\theta)|_{\theta=\theta_0}$ nonsingular, we have $T(x) = \Psi^{-1}\psi(x,\theta)$ and a final recursive step given by $\tilde{\theta} = \hat{\theta} + \sum_i \hat{T}(x_i,\hat{\theta})/n$.

In the likelihood framework, where one maximizes $\log L$ or equivalently minimizes $-\log L$, the NR iteration shown before is effective; if instead of working with the matrix of second derivatives one uses its expectation, the *scoring* method is employed and thus the information matrix now appears in the update step by simply multiplying the expectation by minus one,

$$\theta^* = \hat{\theta} + \mathrm{IM}^{-1}(\hat{\theta})\partial \log L(\hat{\theta}). \tag{17}$$

Alternatively, the BHHH algorithm step given in (14) could be used. For $\hat{\theta}$ consistently estimated in the above formulations, with just one iteration an estimator with the same asymptotic distribution as the ML estimator can be found; therefore, asymptotically efficient estimates are achieved.

5. *M*-ESTIMATION APPROACH

Consider the choice of the following error criterion function, in the light of the previously introduced *M*-estimation approach: $E_n(\hat{\theta}) = \sum_{i=1}^{n} \rho(y_i - \hat{y}_i)$, where $\hat{y}_i$ is computed by the net as a superposition of functions of the input x and the weights θ, as explained before.

The ρ function is a statistically robust function if it satisfies certain properties. Basically, it has either to trim out the abnormal data points or allow for a smoothly descending-to-0 influence function, and still preserve efficiency in estimating the bulk of the observations distribution. It means that it must be fully informative about the middle region of the data distribution, that is, efficient at the Gaussian model, but also able to account for the possible thick tails of the distribution. Thus, $\psi(\cdot) = \rho'(\cdot)$ is the IF to be analyzed and we require it to be bounded and continuous. Some well-known functions offer useful examples.

Example 4 (Huber's minimax).

$$\rho_\lambda(x) = \begin{cases} \dfrac{x^2}{2\lambda} + \dfrac{\lambda}{2} & |x| \leq \lambda, \\ |x| & |x| > \lambda, \end{cases}$$
$$\psi_\lambda(x) = \begin{cases} \dfrac{x}{\lambda} & \text{when } |x| \leq \lambda, \\ \operatorname{sign}(x) & |x| > \lambda, \end{cases} \tag{18}$$

with λ appropriately chosen. One can also have a class of estimators whose $\mathrm{IF} \to 0$ according to a prestablished threshold, that is, $x \geq c$;

Example 5 (Hampel's skipped means).

$$\rho(x,\alpha) = \begin{cases} x^2 & |x| < \sqrt{\alpha}, \\ \alpha & \text{otherwise}, \end{cases}$$
$$\psi(x,\alpha) = \begin{cases} 2x & |x| < \sqrt{\alpha}, \\ 0 & \text{otherwise}. \end{cases} \tag{19}$$

Another robust class of estimators is shown in the following example.

Example 6 (Lorentzian error functions).

$$\rho(x,\sigma) = \log\left(1 + \frac{1}{2}\left(\frac{x}{\sigma}\right)^2\right),$$
$$\psi(x,\sigma) = \frac{2x}{2\sigma^2 + x^2}, \tag{20}$$

where σ is a scale parameter that regulates the slope of the error function shape and thus the degree of decay to zero of the relative *redescending* IF. The correspondent criterion error, for the scale parameter set at $\sigma = 1$, is called least mean log squares in [13].

Instead of minimizing the ρ criterion function, we could equivalently reason in terms of *first-order conditions* (FOC), the equations expressed before through the function ψ. And given that the system of equations that one obtains is nonlinear, we must use the iterative techniques shown before to get solutions. The one-step *M*-estimators are definitely useful in these circumstances since they improve the efficiency of an initial consistent estimator, which in learning problems could be the backpropagation algorithm that runs for some epochs and then is stopped based on a cross-validation method.

We can proceed and generalize the above framework in the following way:

(a) by considering the FOC equation for the local solution θ^* we saw that we must find solution for an equation like $V(\theta) = 0$, where $V(\theta) = \sum_i v(e_i(\hat{\theta})) = \sum_i(\nabla_\theta f(x_i,\hat{\theta})[y_i - f(x_i,\hat{\theta})])$, with $e_i(\cdot)$ the residual computed at pattern i. It would intuitively help, for robustness purposes, to embed the influence function in this context; this step would allow us to merge the *M*-estimation approach directly in the learning paradigm [13]. Then,

(b) by simply using the chain rule one can decompose the error criterion function first derivative such that $\partial\,\mathrm{Loss}(e_i(\theta))/\partial\theta = (\partial\,\mathrm{Loss}(e_i(\theta))/\partial e_i)(\partial e_i/\partial\theta)$ and thus write the original expression equivalently as a function of the influence function; here the IF is playing the role of the weight or influence of the individual residuals on their own first derivatives, that is,

$$\frac{\partial\,\mathrm{Loss}\,(e_i(\theta))}{\partial\theta} = \psi_i \frac{\partial e_i(\theta)}{\partial\theta}. \tag{21}$$

(c) We could extend this strategy by making it work in a recursive fashion, by applying the same principles in the Kalman filter setup; we can indeed exploit the fact that the filter delivers a set of prediction errors, one at each run, and from them we can set a likelihood function equation as in (12). This formula basically corresponds to a functional which we can express in the following compact way: $\Gamma(\hat{\theta}_k) = E[L(\eta, \theta_k)]$, where η is the vector of prediction errors, and thus we can come up with an empirical functional like $(1/n)\sum_{i=1}^{n} L(\eta_i, \theta_k)$, after the filter's runs. In the engineering literature this is known as *prediction error identification* criterion, where the model can also deal with misspecification but still retain the asymptotic properties, in the same spirit of the QML criterion [14].

(d) In order to exploit the IF in such a recursive framework, we note that the same functional Γ numerically optimized in the state space framework can be treated in a recursive way when a SA algorithm is adapted to include it in its updating step, that is, $\hat{\theta}_{n+1} = \hat{\theta}_n + a_n W_n(\eta_n)$, where the W_n term is going to modulate (or equivalently measure the influence of) the impact of the likelihood prediction errors, a sort of analog of the gain factor in the Kalman filter update equation, as already shown in the previous part of this work.

By just allowing for the W_n matrix to be decomposed as in (21), we have the IF entering the update step. The W_n is thus defined as the factorization between the ratio of the first derivatives of the error (or loss) function computed with respect to the prediction errors derivatives and the ratio of the prediction errors derivatives calculated with respect to the weights derivatives. In this way we are considering the influence of possible outliers in the data set onto the prediction errors derivatives. When instead W_n is a composition of a gradient function with a weighting matrix, we are in the case of the modified RM of before, or equivalently the IKF. In the learning fashion we interpret this step like a generalized BP algorithm where the W_n can again be defined as the factorization described before.

6. CONCLUSIONS

From the statistical inference perspective it is important to have the possibility of adopting estimators that use likelihood information in order to reach better asymptotic properties for the final estimates. Equivalently, for neural networks learning processes, it would be ideal to be able to characterize the estimates not only in terms of consistency but also of efficiency. Depending on the error function we may use the information coming from the influence function, by letting it to enter the recursive steps characterizing the Kalman filter or the stochastic approximation algorithms, and thus the generalized backpropagation, so to modulate the impact of deviations from the assumed model on the estimates or by using it to correct for efficiency an initial robust estimate via the one-step Newton-Raphson type formula.

APPENDIX

In (12) we derived the PED likelihood function equation, $\log L = \sum_{t=1}^{N} l_t$, where for univariate time series

$$l_t = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log|D_t| - \frac{1}{2}\eta_t' D_t^{-1}\eta_t. \qquad (22)$$

We differentiate l_t with respect to the jth element of the parameter vector θ (we use two rules for symmetric matrices: $(\partial|X|/\partial t) = |X|\operatorname{tr}[X^{-1}(\partial X/\partial t)]$, $\partial X^{-1}/\partial t = -X^{-1}(\partial X/\partial t)X^{-1}$):

$$\begin{aligned}\frac{\partial l_t}{\partial\theta_j} = &-\frac{1}{2}\operatorname{tr}\left[D_t^{-1}\frac{\partial D_t}{\partial\theta_j}\right] \\ &-\frac{1}{2}\left[\frac{\partial\eta_t'}{\partial\theta_j}D_t^{-1}\eta_t - \eta_t' D_t^{-1}\right. \\ &\qquad\left.\times\frac{\partial D_t}{\partial\theta_j}D_t^{-1}\eta_t + \eta_t' D_t^{-1}\frac{\partial\eta_t}{\partial\theta_j}\right].\end{aligned} \qquad (23)$$

We can take the trace of the terms appearing in the second squared parenthesis and rewrite the expression as follows:

$$\frac{\partial l_t}{\partial\theta_j} = -\frac{1}{2}\operatorname{tr}\left[\left(D_t^{-1}\frac{\partial D_t}{\partial\theta_j}\right)\left(I - D_t^{-1}\eta_t\eta_t'\right)\right] - \left(\frac{\partial\eta_t}{\partial\theta_j}\right)' D_t^{-1}\eta_t. \qquad (24)$$

We now consider (24) and differentiate it with respect to the rth element of the parameter vector θ, so that we obtain the (jr)th element of $\mathrm{IM}(\theta)$:

$$\begin{aligned}\frac{\partial^2 l_t}{\partial\theta_j\partial\theta_r} = &-\frac{1}{2}\operatorname{tr}\left[\frac{\partial\left(D_t^{-1}(\partial D_t/\partial\theta_j)\right)}{\partial\theta_r}\right]\left[I - D_t^{-1}\eta_t\eta_t'\right] \\ &-\frac{1}{2}\operatorname{tr}\left[D_t^{-1}\frac{\partial D_t}{\partial\theta_j}D_t^{-1}\frac{\partial D_t}{\partial\theta_r}D_t^{-1}\eta_t\eta_t'\right] \\ &+\frac{1}{2}\operatorname{tr}\left[D_t^{-1}\frac{\partial D_t}{\partial\theta_j}D_t^{-1}\left(\frac{\partial\eta_t}{\partial\theta_r}\eta_t' + \eta_t\frac{\partial\eta_t'}{\partial\theta_r}\right)\right] \\ &-\frac{\partial^2\eta_t'}{\partial\theta_j\partial\theta_r}D_t^{-1}\eta_t - \frac{\partial\eta_t'}{\partial\theta_j}\frac{\partial D_t^{-1}}{\partial\theta_r}\eta_t \\ &-\frac{\partial\eta_t'}{\partial\theta_j}D_t^{-1}\frac{\partial\eta_t}{\partial\theta_r}.\end{aligned} \qquad (25)$$

By applying the *law of iterated expectations* we take expectations in the above equations conditional on the information available up to time $t-1$; thus, with innovations involved, many of these terms result 0 by definition, and we can find a much simpler $\mathrm{IM}\theta$ expression. We have that $E_z[E_{y/z}(y/z)] = E_y(y)$, or in other terms the random variable $E(y/z)$ has the same expectation as the y one; thus, when z represent previous observations, time series too are accounted for. In our setup this means that since $\eta_t = y_t - \hat{y}_t = y_t - E_{t-1}(y_t)$, we can express the innovation derivatives as functions of the last expression, that is, dependent on the $E_{t-1}(y_t)$ factor $\partial\eta_t/\partial\theta_j = -(\partial/\partial\theta_j)E_{t-1}(y_t)$. Thus, $E_{t-1}((\partial\eta_t'/\partial\theta_j)\eta_t) = (\partial\eta_t'/\partial\theta_j)E_{t-1}(\eta_t) = 0$, given that $E_{t-1}(\eta_t) = 0$; it is the case that $E_{t-1}((\partial^2\eta_t'/\partial\theta_j\partial\theta_r)D_t^{-1}\eta_t) = (\partial^2\eta_t'/\partial\theta_j\partial\theta_r) \times D_t^{-1}E_{t-1}(\eta_t) = 0$, for the same reason of the previous case. The first term in the first row of the master expression too disappears, since $E_{t-1}(\eta_t\eta_t') = D_t$ and thus the last squared parenthesis simplifies to 0. Then, we are left with a simplified second term plus the very last one in the expression; all the other terms drop out.

Thus, the final expression for the (jr)th element of $\mathrm{IM}(\theta)$ is given by

$$\mathrm{IM}_{(j,r)}(\theta) = \frac{1}{2}\sum_t \operatorname{tr}\left[D_t^{-1}\frac{\partial D_t}{\partial \theta_j}D_t^{-1}\frac{\partial D_t}{\partial \theta_r}\right] + E\left[\sum_t \left(\frac{\partial \eta_t}{\partial \theta_j}\right)' D_t^{-1}\frac{\partial \eta_t}{\partial \theta_r}\right], \tag{26}$$

where the $\sum_t$ now appear by the definition

$$-E\left[\frac{\partial^2 \log L}{\partial \theta_j \partial \theta_r}\right] = -E\left[\sum_t \frac{\partial^2 l_t}{\partial \theta_j \partial \theta_r}\right] \tag{27}$$

and the E left in the last term has, asymptotically, a negligible impact on the argument's order of magnitude.

7. ACKNOWLEDGEMENT

The author is a 2001/02 recipient of the ERCIM research fellowship, at CWI, Amsterdam (NL).

REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statistics*, vol. 22, pp. 400–407, 1951.
[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds., vol. 1, pp. 318–362. MIT Press, Cambrige, 1986.
[3] C. M. Kuan and H. White, "Artificial neural networks: an econometric perspective," *Econometric Rev.*, vol. 13, no. 1, pp. 143–143, 1994.
[4] G. R. Chen and H. Oegmen, "Modified extended Kalman filtering for supervised learning," *Internat. J. Systems Sci.*, vol. 24, no. 6, pp. 1207–1214, 1993.
[5] S. Shah and F. Palmieri, "Mekaa fast, local algorithm for training feed-forward neural networks," in *IJCNN*, San Diego, California, 1990, vol. 3, pp. 41–46.
[6] S. Singhal and L. Wu, "Training feed-forward networks with the extended Kalman filter," in *Proc. IEEE ICASSP'89*, Glasgow, Scotland, 1989, pp. 1187–1190.
[7] B. M. Bell and F. W. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Trans. Automat. Control*, vol. 38, no. 2, pp. 294–297, 1993.
[8] E. Capobianco, "A unifying view of stochastic approximation, kalman filter and backpropagation," *IEEE Neural Networks for Signal Processing V*, vol. 87–94, 1995.
[9] F. C. Schweppe, "Evaluation of likelihood functions for gaussian signals," *IEEE Trans. Information Theory*, vol. IT-11, pp. 61–70, 1965.
[10] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge, 1989.
[11] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: The Approach Based on Influence Functions*, John Wiley & Sons, New York, 1986.
[12] P. J. Huber, *Robust Statistics*, John Wiley & Sons, New York, 1981.
[13] K. Liano, "Robust error measure for supervised neural network learning with outliers," *IEEE Transactions on Neural Networks*, vol. 7, no. 1, pp. 246–250, 1996.
[14] L. Ljung, G. Pflug, and H. Walk, *Stochastic Approximation and Optimization of Random Systems*, Birkhäuser Verlag, Basel, 1992.

Enrico Capobianco received Ph.D. in statistical sciences in 1995, from University of Padua (IT), currently an ERCIM Research Fellow at CWI, Amsterdam (NL), doing research with the probability, networks and algorithms cluster about stochastics, wavelets and signals, financial mathematics and time series. Previously conducting research at various institutions, among which Stanford University, from 1994 to 1998, with applied and theoretical work in artificial learning and computational statistics.