33 Robust Fitting by Nonlinear Neural Units

Erkki Oja and Liuyue Wang

A central problem in statistics is fitting a model which is linear in the parameters to a set of observation points. Examples are regression, curve fitting, time series modelling, digital filtering, system theory, and automatic control. The usual approaches are least squares (LS) or total least squares (TLS) regression. The difference between these approaches is shown in Fig. 52 in a simple line fitting example.

The TLS criterion is mathematically equivalent to finding the minor component of the input points, based on the eigenvector of the input covariance matrix corresponding to the smallest eigenvalue. In impulsive and colored noise environments, or in the presence of outliers, these methods are not optimal, however. Then robust fitting, based on a non-quadratic criterion, may give better results than the usual TLS.

The main objection to the use of robust fitting in practice has been a computational one: while the TLS criterion can be solved in closed form and the minor eigenvector can be computed with standard numerical techniques like the singular value decomposition (SVD), this is no longer true for more complicated criterion functions. An iterative gradient descent algorithm is necessary. Neural networks can be an advantage here [1,2,3].

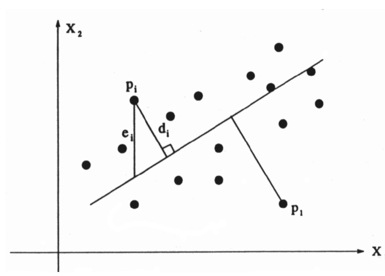

Figure 52: To fit a set of data points by a line, LS minimizes the sum of the squared lengths of the vertical distances e_i, whereas TLS minimizes the sum of the squared lengths of the distances d_i perpendicular to the estimated line.

Referring to Fig. 52, it holds $d_i = |\mathbf{w}^T\mathbf{x}_i|/\|\mathbf{w}\|$. Instead of using the TLS criterion, one may use an alternative criterion by minimizing the sum of certain functions of variable d_i instead of squares:

$$J_f(\mathbf{w}) = 1/N \sum_{i=1}^{N} f(d_i). \tag{93}$$

Generally, function $f(d_i)$ would be a monotonically increasing function of its nonnegative argument d_i. A meaningful choice is an even function, increasing slower than d_i^2. This will decrease the effect of strong outliers on the solution.

Replacing the finite sum in eq. (93) by the theoretical expectation of $f(\mathbf{w}^T\mathbf{x})$ and using a Lagrange multiplier for the constraint $\mathbf{w}^T\mathbf{w} = 1$ gives the following cost function:

$$J_L(\mathbf{w}, \lambda) = E\{f(\mathbf{w}^T\mathbf{x})\} + 1/2\lambda(1 - \mathbf{w}^T\mathbf{w}) \tag{94}$$

whose solution by an on-line gradient descent algorithm gives the following neural learning rule:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k[g(y_k)\mathbf{x}_k - (g(y_k)y_k + 1 - \mathbf{w}_k^T\mathbf{w}_k)\mathbf{w}_k] \quad (95)$$

where $y_k = \mathbf{w}_k^T\mathbf{x}_k$, $g(y)$ is the derivative of $f(y)$, and α_k is a positive learning rate. An especially suitable function for robust TLS fitting is $f(y) = \frac{1}{\beta}lncosh(\beta y)$, giving the usual sigmoid $g(y_k) = tanh(\beta y_k)$ as the neural network learning function. We call this the *Nonlinear Minor Component Analysis (NMCA)* algorithm.

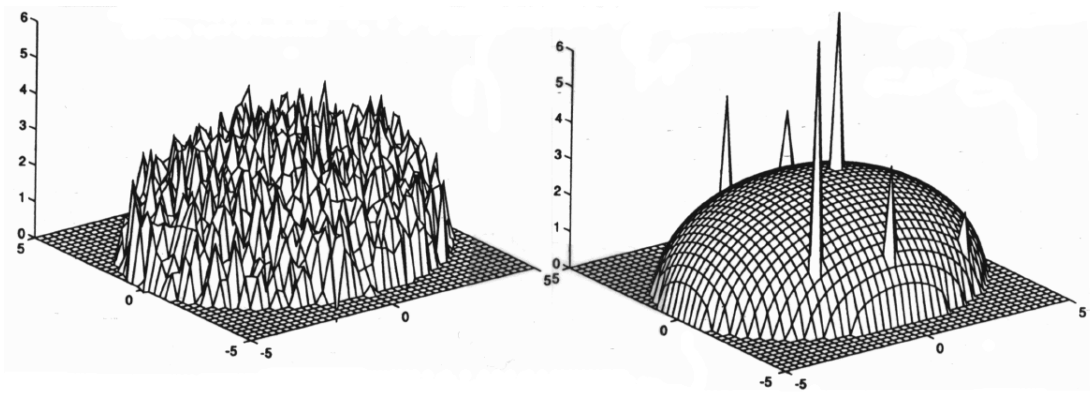

Figure 53: An experiment of surface fitting. Left: Gaussian noise. Right: outliers.

In an experiment of surface fitting [2], a data set D_x was used

$$D_x = \{(x_i, y_i, z_i), i = 1, ..., 993)\}$$

coming from an ellipsoid

$$0.04x^2 + 0.0625y^2 + 0.1111z^2 = 1.$$

Gaussian noise or six strong outliers were added to the sample points, as shown in Fig. 53. Like in line fitting, the problem is now to fit a parameterized model $w_1x^2 + w_2y^2 + w_3z^2 = 1$ to the point set D_x by estimating the parameters w_1, w_2, w_3. The results indicate that the error in the estimated parameters using the Nonlinear MCA algorithm (95) was about one third of the error obtained with conventional LS estimation in the Gaussian noise case and about 6 per cent in the case of outliers.

References

[1] L. Xu, E. Oja, and C. Y. Suen. Modified Hebbian Learning for Curve and Surface Fitting. *Neural Networks 5*, pp. 441–457 (1992).

[2] Oja, E. and Wang, L.: Robust fitting by nonlinear neural units. *Neural Networks 9*, pp. 435 - 444 (1996).

[3] Oja, E. and Wang, L.: Neural fitting: robustness by anti-Hebbian learning. *Neurocomputing 12*, pp. 155 - 170 (1996).