ELSEVIER

Neurocomputing 30 (2000) 23–30

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Structured neural network modelling of multi-valued functions for wind vector retrieval from satellite scatterometer measurements

David J. Evans*,[1], Dan Cornford, Ian T. Nabney

Neural Computing Research Group, Aston Triangle, Aston University, Birmingham B4 7ET, UK

Received 30 September 1998; revised 13 February 1999; accepted 10 March 1999

Abstract

A conventional neural network approach to regression problems approximates the conditional mean of the output vector. For mappings which are multi-valued this approach breaks down, since the average of two solutions is not necessarily a valid solution. In this article mixture density networks, a principled method for modelling conditional probability density functions, are applied to retrieving Cartesian wind vector components from satellite scatterometer data. A hybrid mixture density network is implemented to incorporate prior knowledge of the predominantly bimodal function branches. An advantage of a fully probabilistic model is that more sophisticated and principled methods can be used to resolve ambiguities.

Keywords: Wind vector retrieval; ERS-1 satellite; Probabilistic models; Mixture density networks; Neural networks

1. Introduction

Scatterometers carried on board satellites allow the inference of local wind vectors over the ocean surface [7]. There are two approaches to retrieving local wind vectors [6, this issue], (u, v), from local scatterometer observations, $\boldsymbol{\sigma}^o$, using either a local empirical *forward* or a local empirical *inverse* model. The forward model [11] and

* Corresponding author.

E-mail address: evansdj@aston.ac.uk (D.J. Evans)

[1] Supported by a studentship from the Engineering and Physical Sciences Research Council.

PII: S0925-2312(99)00138-1

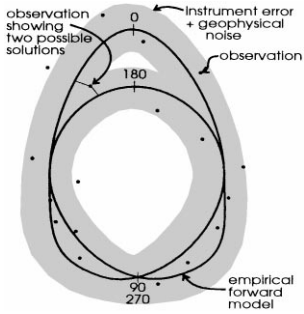


Fig. 1. A two-dimensional sketch of the scatterometer measurement space. The two-dimensional slice is taken through the measurement manifold at constant wind speed. For a noisy observation there are at least two solutions in wind direction.

[8, this issue], which maps $(u, v) \rightarrow \boldsymbol{\sigma}^o$, requires some local inversion to obtain the wind vectors. The current operational method inverts the forward model by finding an estimated $\boldsymbol{\sigma}^o$ on the forward model manifold that is closest to the observed scatterometer measurement [12]. The alternative approach, addressed in this paper, is to directly infer wind vectors from scatterometer data. Models of this form, mapping $\boldsymbol{\sigma}^o \rightarrow (u, v)$, are called *inverse* models. Once the local wind vectors have been inferred, either by using the forward or inverse models, a spatial prior model can be used to infer the wind field over the ocean surface [6].

The scatterometer data is collected by the ERS-1 satellite launched in 1991 by the European Space Agency. The satellite sweeps the ocean surface in swathes approximately 500 km wide, sampling 19 cells across the swathe, where the position across the swathe is given by the antenna beam incidence angle. Each cell is approximately 50 km $\times$ 50 km, and so there is some overlapping between cells. The scatterometer has three antennae, in the same plane, pointing in different directions with respect to satellite movement. The antennae measures a triplet, $\boldsymbol{\sigma}^o$, for each cell. The wind vectors, (u, v), are generated by the numerical weather prediction model run at the European Centre for Medium range Weather Forecasting.

Previous work [4] has shown that there is a unique set of wind vectors called the *noisy ambiguity set* which is identifiable from a single scatterometer measurement; that is, the inverse mapping exists and is multi-valued. The multi-valued nature of the inverse mapping arises largely from noise on the observations. This is illustrated in Fig. 1, a sketch of a two-dimensional slice through the three-dimensional measurement space, at a fixed incidence angle. The position of the observation on the model manifold is a function of wind speed and direction [11]. A noisy observation is unlikely to lie on the model manifold, making it uncertain from which of the two model branches the observation originates. Thus, there are typically at least two solutions for wind direction from a single scatterometer observation. These two solutions are roughly 180° apart in direction, and are generally referred to as the ambiguous solutions [12].

1.1. Background

Neural networks have been applied to wind retrieval from scatterometer observations. In [13] neural networks were used to infer wind direction and speed directly from simulated scatterometer data. For each incidence angle, the model consisted of two feed-forward neural networks. One network modelled wind speed with a conventional regression approach, the other modelled wind direction by classifying it into 36 bins representing 10° intervals. The inputs to the neural network took included information from the surrounding cells, giving a spatial context. In addition to the scatterometer data, the wind direction network also took wind speed as an input. Simulated data was used because ERS-1 was not operational at that time. The results showed neural networks to be a promising avenue of investigation for a solution to this inverse problem. In [9] the models of Thiria et al. [13] are trained using data collected from ERS-1. Performance of the models in [9] is shown to improve upon results obtained by the operational wind retrieval system at the European Space Agency. Inclusion of a spatial context means that the network also carries out some disambiguation (improving its accuracy on individual cells). However, for reasons discussed in [6], such a model cannot be used in a general disambiguation procedure since it is not purely local.

In [3] wind speed was modelled using a multi-layer perceptron while the wind direction was modelled by a mixture density network with circular normal kernel densities [2] to model the full conditional probability density of the wind direction given the scatterometer measurements. In addition to the scatterometer measurements, the incidence angle of the mid-beam antenna was included as an input to the networks. The wind-speed model performed within the designed specification of the instrument of 2 m s^{-1}. For wind direction, the models learned the inherent ambiguity in the problem, but did not perform as well as the models of Richaume et al. [9].

In [10] it is shown that it is preferable to analyse wind-vector components in Cartesian coordinates rather than wind speed and direction (polar coordinates), as the noise distribution on the the predicted wind-vector components is shown to be spherically Gaussian. In this paper, we use this information and directly model the Cartesian wind-vector components from scatterometer observations for the first time.

2. Modelling multi-valued functions

2.1. Theory of mixture density networks

Mixture density networks (MDNs) provide a framework for modelling conditional probability density functions, denoted by $P(\boldsymbol{t}|\boldsymbol{x})$ [5,1]. The distribution of the outputs, $\boldsymbol{t} \in \mathbb{R}^c$, is described by a parametric model whose parameters, $\boldsymbol{Z}$, are determined by the output of a neural network, which takes $\boldsymbol{x}$ as its inputs. The general model is described by

$$P(\boldsymbol{t}|\boldsymbol{x}) = \sum_{j=1}^{M} \alpha_j(\boldsymbol{x})\phi_j(\boldsymbol{t}|\boldsymbol{x}), \tag{1}$$

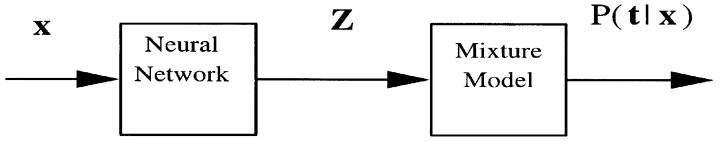


Fig. 2. The structure of a mixture density network. The inputs $\boldsymbol{x}$ are feed through the neural network. The outputs of the neural network, $\boldsymbol{Z}$, define the parameters of the Gaussian mixture model.

and

$$\sum_{j=1}^{M} \alpha_j(\boldsymbol{x}) = 1, \tag{2}$$

where $\alpha_j(\boldsymbol{x})$ represents the mixing coefficients (which depend on $\boldsymbol{x}$), $\phi_j(\boldsymbol{t}|\boldsymbol{x})$ are the kernel distributions of the mixture model (whose parameters also depend on $\boldsymbol{x}$) and M is the number of kernels in the mixture model. Generally, the kernels used are c-dimensional spherical Gaussians of the form

$$\phi_j(\boldsymbol{t}|\boldsymbol{x}) = \frac{1}{(2\pi)^{c/2}\sigma_j^c(\boldsymbol{x})} \exp\left(-\frac{\|\boldsymbol{t} - \boldsymbol{\mu}_j(\boldsymbol{x})\|^2}{2\sigma_j^2(\boldsymbol{x})} \right). \tag{3}$$

In principle a Gaussian mixture model with sufficiently many kernels of the type given by Eq. (3) can approximate any density function providing the parameters are chosen correctly [5]. It follows then that for any given value of $\boldsymbol{x}$, the mixture model (1) can model the conditional density function $P(\boldsymbol{t}|\boldsymbol{x})$. To achieve this the parameters of the mixture model are taken to be general continuous functions of $\boldsymbol{x}$. The output of the neural network is a vector, $\boldsymbol{Z}$, which contains the parameters that define the coefficients of the mixture model conditional on the inputs $\boldsymbol{x}$. For spherical Gaussian mixture models the coefficients are, α_j the mixing coefficient for the jth kernel, μ_{jk} the kth element of the centre of the jth kernel and σ_j^2 the width or variance of the jth kernel. The parameter vector, $\boldsymbol{Z}$, is summarised as

$$\boldsymbol{Z} = [\underbrace{\alpha_1, \alpha_2, \ldots, \alpha_M}_{M \text{ mixing coefficients}}, \underbrace{\boldsymbol{\mu}_1^{\mathrm{T}}, \boldsymbol{\mu}_2^{\mathrm{T}}, \ldots, \boldsymbol{\mu}_M^{\mathrm{T}}}_{M \text{ kernel centres}}, \underbrace{\sigma_1^2, \sigma_2^2, \ldots, \sigma_M^2}_{M \text{ kernel widths}}].$$

It is this combination of a Gaussian mixture model, whose parameters are dependent on the output of feed forward neural network that takes $\boldsymbol{x}$ as its inputs, that is referred to as a *mixture density network* and is represented schematically in Fig. 2.

By choosing sufficient kernels in the mixture model and a neural network with sufficient hidden units the MDN can approximate as closely as desired any conditional density, $P(\boldsymbol{t}|\boldsymbol{x})$ [1]. The neural network element of the MDN is implemented with a standard multi-layer perceptron (MLP) with single hidden layer of tanh units and an output layer of linear units.

2.2. Modelling the geophysical problem

In the context of this application each input pattern for the MDN, $\boldsymbol{x}$, is the observed scatterometer data, $\boldsymbol{\sigma}^o$ and the cosine of the incidence angle, θ. Modelling the wind vector components directly implies that the targets of the MDN, $\boldsymbol{t}$, are the wind-vector components (u, v). The general description of the MDN, (1), is then re-expressed using geophysical parameters as

$$P(u, v|\boldsymbol{\sigma}^o, \theta) = \sum_{j=1}^{M} \alpha_j(\boldsymbol{\sigma}^o, \theta)\phi_j(u, v|\boldsymbol{\sigma}^o, \theta). \tag{4}$$

2.3. Modelling the inherent geophysical knowledge

We also investigated a hybrid architecture, which is a modification of the standard MDN architecture, in order to model the known geophysical knowledge of the problem, the 180° ambiguity in wind direction. The hybrid MDN has two kernels. One kernel is free to move, and the other is positioned diametrically opposite the first in (u, v) space, by taking the negative mean of the free moving kernel. The simplified model becomes

$$P(u, v|\boldsymbol{\sigma}^o, \theta) = \alpha(\boldsymbol{\sigma}^o, \theta)\phi(u\, v|\boldsymbol{\sigma}^o, \theta) + (1 - \alpha(\boldsymbol{\sigma}^o, \theta))\psi(u, v|\boldsymbol{\sigma}^o, \theta), \tag{5}$$

where the kernels are defined by diametrically opposed spherical Gaussians with common variances:

$$\phi(u, v|\boldsymbol{\sigma}^o, \theta) = \frac{1}{2\pi\sigma^2(\boldsymbol{\sigma}^o, \theta)} \exp\left(- \frac{||(u, v) - \boldsymbol{\mu}(\boldsymbol{\sigma}^o, \theta)||^2}{2\sigma^2(\boldsymbol{\sigma}^o, \theta)} \right), \tag{6}$$

$$\psi(u, v|\boldsymbol{\sigma}^o, \theta) = \frac{1}{2\pi\sigma^2(\boldsymbol{\sigma}^o, \theta)} \exp\left(- \frac{||(u, v) + \boldsymbol{\mu}(\boldsymbol{\sigma}^o, \theta)||^2}{2\sigma^2(\boldsymbol{\sigma}^o, \theta)} \right). \tag{7}$$

3. Results

In total 12 networks were trained,[2] using early stopping for regularisation. The performance of the networks is evaluated using the vector root mean square (RMS) error between the predicted and target values on a test data set and the percentage of predicted directions from the two most probable modes that fall within 20° of the target wind direction. The results are summarised in Table 1.

The results suggest that model performance is more sensitive to the number of kernels in the MDN configuration than the number of hidden units in the MLP.

[2] When training the MDNs, the inputs are assumed to be noiseless in comparison to the noise on the targets.

Table 1
Results of the 12 MDN configurations. These results are generated from a test data set of 5000 examples

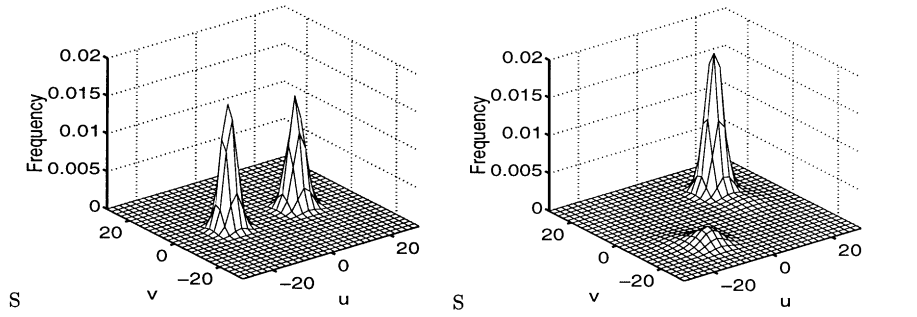
MDN architecture Kernels	Hidden Units	Vector RMS errors	Percentage within 20°
2 (Hybrid)	35	4.33	73.38
2 (Hybrid)	50	4.18	70.32
2	35	4.02	72.76
2	50	4.03	74.10
4	20	3.82	76.76
4	25	3.69	76.82
4	30	3.90	76.64
4	35	3.89	76.94
4	50	3.73	77.12
4	90	4.29	76.64
12	35	4.58	76.74
12	50	4.24	77.16

Recent work using a specially selected ERS-2 data set, which has outliers removed by hand, has shown a significant improvement in our model statistics. Our best model (computed on the new ERS-2 test set, which is independent of the training and validation set), with 25 units in the hidden layer of the MLP and four kernels in the mixture model, has a vector RMS error of 2.33 m s^{-1} and percentage within 20° of 85.07.

3.1. Discussion

The complexity of the mapping, $(\boldsymbol{\sigma}^o, \theta) \rightarrow (u, v)$, is modelled by the MLP part of the MDN. The focus of the investigation is on MDNs with four kernels. Here the difference in the performance of percentage within 20° between the best and worst model is less than 0.5%, and for vector RMS error is 0.6 m s^{-1}. It seems possible these differences are due to different initial positions on the error surface. The model with 90 hidden units does not perform as well for vector RMS error, and it is suggested that this is due to the model over-fitting. The model with 20 hidden units gives a good indication of the complexity of the mapping, $(\boldsymbol{\sigma}^o, \theta) \rightarrow (u, v)$.

Comparing the hybrid MDNs with the MDNs with two kernels it is interesting to note that the directional performance is similar, and the vector RMS differs by less than 0.3 m s^{-1} between best and worst case. This gives strong evidence to suggest the solution is predominantly bimodal (see Fig. 3) with these modes being approximately 180° apart in direction. However, the models with four kernels out-perform the models with two. The complexity of the density model in the MDN is related to the number of kernels in the Gaussian mixture model. The improved performance of the MDNs with four kernels is attributed to two factors. Firstly, although the results



(a) Strongly bimodal.

(b) Bimodal, but one mode dominating.

Fig. 3. The conditional probability distribution of the wind vectors (u,v) given the scatterometer data for a MDN with four kernels, 20 hidden units, on two different input patterns.

suggest that the modes of the conditional distribution are predominantly bimodal, they are not always Gaussian or spherically symmetric, suggesting that the noise on the targets is heavier tailed than originally assumed. Four kernels in the MDN are able to model the non-Gaussian, non-spherical modes in the conditional probability distribution more efficiently than two kernels. Secondly, the increased flexibility of four kernels permits the MDN to place kernels into four quadrants of (u,v) when appropriate, flexibility which is not available to the models with two kernels. Two further experiments obtained results for MDNs with 12 kernels. The results show that there is an increase in the vector RMS error. This is due to the model over-fitting, since it has sufficient flexibility to model both the *underlying data generator* and the noise on the training data set, and hence yields poor results for the test set.

Bench marking against previous work is difficult because of the different data sets used when training and testing the models. However, bearing this in mind, the networks in [3], which model each cell independently, achieved a correct solution within 20° roughly 73% of the time when considering the two most probable solutions. The results reported in [9], achieve a correct solution more than 85% of the time in wind direction when considering the two most probable solutions. However in [9] it must be noted that spatial information is also provided at the inputs to the networks, which provides additional disambiguation skill [3].

The results of this study improved on the results of the local models in [3]. When the local models trained in this study are applied using the methods proposed in [6, this issue] it is hoped that we can further improve performance.

4. Conclusions

In this paper a novel method for modelling the Cartesian wind-vector components, (u, v), directly from scatterometer data has been introduced. By using the MDN

framework, a fully probabilistic model, $P(u, v|\boldsymbol{\sigma}^o, \theta)$, has been developed which describes the joint probability distribution of the wind vectors given the scatterometer observations. The hybrid MDN has shown that the solution is predominantly bimodal, agreeing with earlier work [12]. Training MDNs with several different architectures suggests that these are the best results achievable, given the data, by local modelling of the inverse mapping $(\boldsymbol{\sigma}^o, \theta) \rightarrow (u, v)$.

On going work, using improved data selection techniques for generating training data sets from the recently available ERS-2 satellite data, has shown significant improvements in model performance.

Acknowledgements

This work is supported by the European Union funded NEUROSAT programme (grant number ENV4 CT96-0314).

References

[1] C.M. Bishop, Neural Networks and Pattern Recognition, Oxford University Press, Oxford, 1995
[2] C.M. Bishop, I.T. Nabney, Modelling conditional probability distributions for periodic variables, Neural Comput. 8 (1996) 1123–1133.
[3] D. Cornford, I.T. Nabney, C.M. Bishop, Neural network based wind vector retrieval from satellite scatterometer data Neural Comput. Appl., in press.
[4] D. Long, J.M. Mendel, Identifiability in wind estimation from scatterometer measurements, IEEE Trans. Geosci. Remote Sensing 29 (1991) 268–276.
[5] G.J. McLachlan, K.E. Bashford, Mixture Models: Inference and Applications to Clustering, Marcel Dekker, New York, 1988.
[6] I.T. Nabney, D. Cornford, C.K.I. Williams, Bayesian inference for wind field retrieval, Neurocomputing, this issue.
[7] D. Offiler, The calibration of ERS-1 satellite scatterometer winds, J. Atmos. Ocean. Technol. 11 (1994) 1002–1017.
[8] D. Cornford, G. Ramage, I.T. Nabney, A scatterometer neural network sensor model with input noise, Neurocomputing, this issue.
[9] P. Richaume, F. Badran, M. Crepon, C. Mejia, H. Roquet, S. Thiria, Neural network wind retrieval from ERS-1 scatterometer data, J. Geophys. Res., in press.
[10] A. Stoffelen, Toward the true near surface wind speed: error modeling and calibration using triple location, J. Geophys. Res. 103 (1998) 7755–7766.
[11] A. Stoffelen, D. Anderson, Scatterometer data interpretation: estimation and validation of the transfer function CMOD4, J. Geophys. Res. 102 (1997) 5767–5780.
[12] A. Stoffelen, D. Anderson, Scatterometer data interpretation: measurement space and inversion, J. Atmos. Ocean. Technol. 14 (1997) 1298–1313.
[13] S. Thiria, C. Mejia, F. Badran, A neural network approach for modelling nonlinear transfer functions: application for wind retrieval from spaceborne scatterometer data, J. Geophys. Res. 98 (1993) 22 827–22 841.