Fischer, Manfred M.

Conference Paper

Principles of Neural Spatial Interaction Modeling

43rd Congress of the European Regional Science Association: "Peripheries, Centres, and Spatial Development in the New Europe", 27th - 30th August 2003, Jyväskylä, Finland

Provided in Cooperation with:
European Regional Science Association (ERSA)

Suggested Citation: Fischer, Manfred M. (2003) : Principles of Neural Spatial Interaction Modeling, 43rd Congress of the European Regional Science Association: "Peripheries, Centres, and Spatial Development in the New Europe", 27th - 30th August 2003, Jyväskylä, Finland, European Regional Science Association (ERSA), Louvain-la-Neuve

This Version is available at:
https://hdl.handle.net/10419/116239

Revised version 30-04-2003

CHAPTER XX

Principles of Neural Spatial Interaction Modelling

MANFRED M. FISCHER

Department of Economic Geography & Geoinformatics
Vienna University of Economics and Business Administration

Abstract

Neural spatial interaction models represent the most recent innovation in the design of spatial interaction models. They are receiving increasing attention in recent years because of their powerful universal approximation properties. In essence they are devices for non-parametric statistical inferences, providing an elegant formalism for spatial interaction modelling. This contribution meets an urgent demand for methodological guidelines on how to develop robust applications that work from a statistical perspective. It introduces various components of a methodology for neural spatial interaction modelling that consists of a model specification framework to produce consistent estimators, a parameter estimation framework to compute parameter estimates that optimize an explicit fitness criterion and a framework to evaluate the model performance.

JEL classification: C31, C45, R19
Keywords: Spatial interaction, neural networks, non-parametric non-linear models, model selection, parameter estimation, model adequacy testing

1 INTRODUCTION

Regional science is a rich discipline at the cross-roads of economics and geography. A closer look at the history of the discipline teaches us that the field of spatial interaction has a long and deep intellectual tradition[1]. That there have been relatively few papers in this field in recent years is merely a function of the hiatus that followed a very active period of theory development in the 1960s and 1970s, the heady days of Stewart and Warntz, Stouffer, Isard, Wilson and Alonso. The empiricism that emanated from their theoretical and methodological contributions filled regional science and geography journals. The lull came not so much because interest decreased, but very little theoretical progress has been achieved. One exception was the excitement over the work of Fotheringham on competing destinations in the early 1980s when several new models were developed and new perspectives added (Fischer and Getis, 1999).

In more recent years, the major influence stems both from the emerging data-rich environment and from technological innovations. The powerful and fast computing environment now available has brought many scholars to spatial interaction theory once again, either by utilizing evolutionary computation to breed novel forms of spatial interaction models (see Openshaw, 1988; Turton, Openshaw and Diplock, 1997) or applying neural network theory to spatial interaction, first proposed by Fischer and Gopal (1994) and later extended by many others [including Fischer and Leung, 1998; Bergkvist, 2000; Reggiani and Tritapepe, 2000; Mozolin, Thill and Usery, 2000; Fischer and Reismann, 2002a, b; Fischer, 2000, 2002a, b; Fischer, Reismann and Hlavackova-Schindler, 2003].

The novelty about neural spatial interaction models lies in their ability to model non-linear spatial interaction processes with few – if any – a priori assumptions about the nature of the generating process. A

[1] It is beyond the scope of this contribution to offer a survey of this tradition. The reader is referred to the wealth of historical material in Carrothers (1956), Isard and Bramhall (1956), Olsson (1965), Wilson (1967, 1970), Batten and Boyce (1986), Fotheringham and O'Kelly (1989), Sen and Smith (1995), among others.

major weakness of neural spatial interaction modelling is the lack of established procedures for performing tests of statistical significance for the various model parameters that have been estimated. This is a serious disadvantage in the regional science community where there is a strong culture for testing not only the predictive power of a model or the sensitivity of the dependent variable to changes in the inputs but also the statistical significance of the finding at a specified level of confidence. This contribution meets an urgent demand for methodological guidelines on how to develop robust applications that work from a statistical perspective.

The remainder of this chapter is structured as follows. The next section introduces the class of neural spatial interaction models of interest, and sets forth the context in which spatial interaction modelling will be considered. The sections that follow present constituent components of a methodology for neural spatial interaction modelling that comprises a model selection framework to produce consistent estimators (see Section 3), a parameter estimation framework to compute a set of parameter estimates that optimize an explicit fitness criterion (see Section 4), and a framework to evaluate the model performance (see Section 5). Section 6 concludes the chapter.

2 NEURAL SPATIAL INTERACTION MODELLING AND ASSUMPTIONS

In this contribution we will be concerned with data generated according to the following conditions.

Assumption A: Observed data are the realization of a sequence $\{z_u=(\boldsymbol{x}_u, y_u),\ u=1,\ \ldots,\ U\}$ of independent identically distributed (*iid*) random $(N+1)$x1 vectors, $N \in \mathbb{N}$, with zero mean and constant variance.

The random variables y_u represent bi-locational spatial interaction flows [=targets]; their relationship to the variables $\boldsymbol{x}_u$ [such as origin-specific, destination-specific and separation attributes] is of primary interest.

When $E(y_u)<\infty$, the conditional expectation of y_u given $\boldsymbol{x}_u$ exists, denoted as $g = E(y_u \mid \boldsymbol{x}_u)$. Defining $\varepsilon_u \equiv y_u - g(\boldsymbol{x}_u)$ we can also write

$$y_u = g(\boldsymbol{x}_u) + \varepsilon_u \tag{1}$$

The unknown function $g(\boldsymbol{x}_u)$, called *spatial interaction function*, embodies the systematic part of the stochastic relation between y_u and $\boldsymbol{x}_u$. The error ε_u is noise, with the property $E(\varepsilon_u \mid \boldsymbol{x}_u)=0$ by construction. Knowledge of $E(y_u \mid \boldsymbol{x}_u)$ or of the underlying function $g(\boldsymbol{x}_u)$ is equivalent since $E(\varepsilon_u \mid \boldsymbol{x}_u)=0$. Neural spatial interaction models may be viewed as estimators of the conditional density $E(y_u \mid \boldsymbol{x}_u)$, or in other words the on average realization of y given $\boldsymbol{x}$. They make no a priori assumption regarding the functional form of $g(\boldsymbol{x}_u)$. Thus, they are non-parametric estimators, as opposed to parametric, where a priori assumptions are made.

We are interested in a methodology for the case of *unconstrained* spatial interaction. The objective of such a methodology is to construct an estimator of the unknown spatial interaction function $g(\boldsymbol{x}_u)$, denoted $\Omega(\boldsymbol{x}, \boldsymbol{w})$ where $\boldsymbol{w}$ is a set of p free parameters and $z_u=(\boldsymbol{x}_u, y_u)$ a finite set of observations. A well specified estimator will have the following characteristics:

- it will provide a comfortable fit with the data,
- the expectation $E[\varepsilon \mid \boldsymbol{x}]=0$.

The task of model selection involves to choose a functional form from a number of possibly competing alternatives [the model specification task], and to estimate the parameters in a way which satisfies a fitness criterion [the parameter estimation task].

Our interest in this contribution is focused on the output functions of unconstrained neural spatial interaction models based upon single hidden

layer feedforward networks[2] (see Fischer and Reismann, 2002b). These functions have the following properties:

Assumption B: Model output is given by a function Ω: $\Re$ x $W \rightarrow \Re$ where W is a compact subset of $\Re^p$ (p integer) such that for each $w \in W$ $\Omega(., w)$: $\Re^N \rightarrow \Re$ is (Borel) measurable.

Model parameters [network weights] are here restricted to lie in a compact set W of finite dimension p, where p is the total number of parameters. This requirement for model output functions is satisfied for single hidden layer network models of the form

$$\Omega^H(x, w) = \psi(w_{00} + \sum_{h=1}^{H} w_{0h}\, \varphi(\sum_{n=1}^{N} w_{1hn}\, x_n)) \tag{2}$$

where the N-dimensional euclidean space (in practice generally, $N = 3$) is the input space and the 1-dimensional euclidean space the output space. $x = (x_1, ..., x_N)$ is the input vector that represents variables which characterize the origin and the destination of spatial interaction as well as their separation.

This class of general neural spatial interaction models is a set of network models which share the same architecture and whose individual members are continuously parameterized by the $p=(HN + H + 1)\times 1$-dimensional vector $w^H \equiv (w_0, w_1)$ where $w_0 \equiv (w_{00}, w_{01}, ..., w_{0H})$ contains the hidden to output unit weights and $w_1 \equiv (w_{10}, ..., w_{1H})$ with $w_{1h} \equiv (w_{1h1}, ..., w_{1hN})$ the input to hidden unit weights. $\varphi(.)$ is a hidden layer transfer function, $\psi(.)$ an output unit transfer function, both continuously differentiable of order 2 on $\Re$. The nested form of Equation (2) is the main reason that fitted spatial interaction models are

[2] Although unconstrained neural spatial interaction models of type (2) represent a rich and flexible family of spatial interaction function approximators for real world applications, they may be of little practical value in situations where a priori information is available on accounting constraints on the predicted flows. For such cases Fischer, Reismann and Hlavackova-Schindler (2003) proposed a novel neural approach based on a modular product unit neural network architecture.

so difficult to interpret. The model (2) is explicitly indexed by the number, H, of hidden units in order to indicate the dependence. Without any loss of generality in this contribution we consider only neural spatial interaction models with fixed H. To simplify notation, we, thus, drop the superindex H hereafter.

3 MODEL SPECIFICATION AND THE FITNESS FUNCTION

An important step in the specification of the general neural spatial interaction model (2) is the choice of the transfer functions, $\psi(.)$ and $\varphi(.)$. These can be any non-linear functions as long as they are continuous and differentiable. Typically, they are sigmoidal, the hyperbolic tangent or a thermodynamic-like function. All these functions belong to the family

$$\Gamma = \{\gamma = \gamma(\boldsymbol{x}, r, s, t) \mid \boldsymbol{x}, r \in \Re;\ s, t \in \Re - \{0\}\} \tag{3}$$

where $\gamma(.)$ is defined as follows

$$\gamma(\boldsymbol{x}) = r + s(1 + \exp t\,\boldsymbol{x})^{-1}. \tag{4}$$

When r=s=1 and t=-1 the asymmetric sigmoid is obtained, which is the most commonly used function.

The second and main step in model specification involves the choice of an individual member of the model class. Over the years of neural network development an impressive array of model specification procedures have been proposed. Many tackle the problem of searching over the specification space and parameter space simultaneously. But the most important one is the so-called *discrimination approach* where the individual members of the model class under consideration are evaluated using a fitness criterion that penalizes the in-sample performance of the model, as the complexity of the functional form [that is, H] increases.

The fitness criterion, also known as cost functional, performance or loss function or discrepancy criterion, is generally defined as the average

$$\lambda(\boldsymbol{w}) = \frac{1}{U}\sum_{u=1}^{U}\pi(\boldsymbol{z}_{true},\boldsymbol{w}) \tag{5}$$

where $\boldsymbol{z}_{true}$ stands for the pair $(\boldsymbol{x}, y_{true})$ and $\pi(\boldsymbol{z}_{true}, \boldsymbol{w})$ is a pairwise discrepancy criterion. The superindex *true* emphasizes the fact that y_{true} is different from the observed value y that is used to fit the model. The fitness function should be such that it increases if $g(\boldsymbol{x})$ and $\Omega(\boldsymbol{x}, \boldsymbol{w})$ are considered to become less similar.

The objective of the learning process is to identify a parameter vector, say $\tilde{\boldsymbol{w}}$, which minimizes the loss function (5). Let us assume for the moment that the solution to the minimization problem

$$\tilde{\boldsymbol{w}} = \arg\min\{\lambda(\boldsymbol{w}) \text{ with } \boldsymbol{w} \in \boldsymbol{W}, \boldsymbol{w} \subseteq \Re^{p} \text{ and } p = (HN + H + 1)\} \tag{6}$$

exists and it is unique. We term $\lambda(\tilde{\boldsymbol{w}})$ the *discrepancy due to bias*. It does not depend in any way on the sample size or the sample. In general, $\lambda(\tilde{\boldsymbol{w}})$ decreases with the dimensionality of the parameter vector. The model $\Omega(\boldsymbol{x}, \tilde{\boldsymbol{w}})$ is called the *best approximating model* for the family $\Omega(\boldsymbol{x}, \boldsymbol{w})$ and the loss function $\lambda(\boldsymbol{w})$. Following Zapranis and Refenes (1999) we will term this the *empirical loss*, and will denote it as $\lambda_U(\boldsymbol{w})$, a consistent estimator of $\lambda(\boldsymbol{w})$. A common choice of empirical loss is

$$\lambda_U(\boldsymbol{w}) = \frac{1}{U}\sum_{u=1}^{U}\pi(\boldsymbol{z}_u,\boldsymbol{w}) \tag{7}$$

where $\boldsymbol{z}_u$ belongs to the training sample $S_U=\{(\boldsymbol{z}_u, y_u), u=1, ..., U\}$. It can be shown that as the size of the sample tends to infinity, $\lambda_U(\boldsymbol{w})$ converges to $\lambda(\boldsymbol{w})$ (see White, 1989).

Let us assume that the solution to the following minimization problem

$$\hat{\boldsymbol{w}}_U = \arg\min \{\lambda_U(\boldsymbol{w}) \text{ with } \boldsymbol{w} \in \boldsymbol{W},\ \boldsymbol{w} \subseteq \Re^p\} \tag{8}$$

exists, then $\hat{\boldsymbol{w}}_U$ is called a *minimum discrepancy estimator* of $\tilde{\boldsymbol{w}}$[3]. The discrepancy between the best approximating neural spatial interaction model $\Omega(\boldsymbol{x}, \tilde{\boldsymbol{w}})$ and $\Omega(\boldsymbol{x}, \hat{\boldsymbol{w}}_U)$ is called the *discrepancy due to variance*. It expresses the magnitude of the lack of fit due to sample variation and does depend on the data and on the parameter estimation procedure utilized. Its expectation increases in general as the dimensionality of the parameter vector also increases.

Neither discrepancy due to bias nor discrepancy due to estimation can be computed unless the underlying spatial interaction function $g(\boldsymbol{x})$ is known. Since $g(\boldsymbol{x})$ and consequently $\tilde{\boldsymbol{w}}$ are unknown, the learning process in practice consists of minimizing Equation (8) where $\lambda_U(\boldsymbol{w})$ is given from Equation (7). To calculate the empirical loss with Equation (7) one needs first to define a discrepancy criterion. Although there is no such universally acceptable criterion, the pairwise squared difference between $g(\boldsymbol{x})$ and $\Omega(\boldsymbol{x}, \hat{\boldsymbol{w}})$ is most widely used discrepancy criterion[4]:

$$\pi(\boldsymbol{z}_{true}, \boldsymbol{w}) = \tfrac{1}{2}\left[g(\boldsymbol{x}) - \Omega(\boldsymbol{x}, \hat{\boldsymbol{w}})\right]^2 \tag{9}$$

Note that different data-generating assumptions different from *Assumption A* would lead to different criteria that result in different loss functions. One important example derived from Poisson processes is the Kullback-Leibler loss function (see Kullback and Leibler, 1951)

$$\lambda^{KL}(\boldsymbol{w}) = -E\left[\log p_H(\boldsymbol{x}, \boldsymbol{w})\right] \tag{10}$$

where $p_H(\boldsymbol{x}, \boldsymbol{w})$ is the probability density function of the approximating neural spatial interaction model. The minimum discrepancy estimator associated with this loss function is the maximum likelihood estimator (see Fischer, 2002a).

---

[3] If $\hat{\boldsymbol{w}}_U$ is a minimum discrepancy estimator [i.e. a solution of Equation (8)] then it can be shown that $\hat{\boldsymbol{w}}_U$ asymptotically converges to $\tilde{\boldsymbol{w}}$ (Gallant and White, 1988).

[4] The factor $\frac{1}{2}$ serves the purpose to simplify the formulae for the derivatives of $\lambda_U(\boldsymbol{w})$.

The problem of selecting the appropriate neural spatial interaction model can alternatively be viewed as balancing the bias and variance parts of the expected squared difference between $\Omega(\boldsymbol{x}, \boldsymbol{w})$ and $g(\boldsymbol{x})$. An under-parameterized model will have a large bias and smooth out some of the underlying structure in the data, while one model that has too much flexibility in relation to the particular data set will overfit the data and have a large variance. The model will be very sensitive to the data and characteristically far from $g(\boldsymbol{x})$. Balancing these two opposing forces is far from trivial in practice. Various ways to controlling complexity of a neural spatial interaction model are discussed in Fischer (2000).

4 PARAMETER ESTIMATION AND PROCEDURES

Given the class of neural spatial interaction models (2), parameter estimation simply consists of solving the minimization problem (8), where $\lambda_U(\boldsymbol{w})$ is given by Equation (7) and the discrepancy criterion is usually the [halved] squared differences between targets [y_u] and model forecasts[5]. In this case the resulting empirical loss $\lambda_U(\boldsymbol{w})$ is the ordinary least squares function

$$\lambda_U(\boldsymbol{w}) = \tfrac{1}{2U} \sum_{u=1}^{U} \left[y_u - \Omega(\boldsymbol{x}_u, \boldsymbol{w}) \right]^2 . \tag{11}$$

As above we denote the solution to this minimization problem as $\hat{\boldsymbol{w}}_U$ where the hat signifies that it is an estimator of the parameter vector $\tilde{\boldsymbol{w}}$ and the subscript U emphasizes its dependence on the sample size. Since the loss function $\lambda_U(\boldsymbol{w})$ is a complex non-linear function of $\boldsymbol{w}$, the problem of estimating the model parameters $\hat{\boldsymbol{w}}_U$ by means of optimizing some performance criterion does not have a well-defined closed-form

[5] Because $E[\{y\text{-}g(\boldsymbol{x})\}^2]$ is simply the variance of y given x and does not depend on the data, it follows that by minimizing $E[\{y\text{-}\Omega(\boldsymbol{x}, \boldsymbol{w})\}^2]$ one also minimizes $E[\{g(\boldsymbol{x})\text{-}\Omega(\boldsymbol{x}, \boldsymbol{w})\}^2\}]$ that is the mean squared error of $\Omega(\boldsymbol{x}, \boldsymbol{w})$ and a natural measure of the performance of model $\Omega(\boldsymbol{x}, \boldsymbol{w})$ as a predictor (Zapranis and Refenes, 1999).

solution. But iterative procedures are available for this purpose. Two types of iterative procedures may be distinguished: local search and global search procedures.

Local search procedures characteristically use derivative information of $\lambda_U(\boldsymbol{w})$ within a local iterative process in which an approximation to the function in a neighbourhood of the current point in parameter space is minimized. The general scheme of the iteration process may be characterized as follows (Fischer, 2001b):

(i) choose an initial vector $\boldsymbol{w}$ in parameter space and set τ=1,
(ii) determine a search direction $\boldsymbol{d}(\tau)$ and a step size $\eta(\tau)$ so that

$$\lambda_U(\boldsymbol{w}(\tau)+ \eta(\tau)\, \boldsymbol{d}(\tau)) < \lambda_U(\boldsymbol{w}(\tau)) \qquad \tau=1, 2, ... \tag{12}$$

(iii) update the parameter vector

$$\boldsymbol{w}(\tau+1)= \boldsymbol{w}(\tau)+\eta(\tau)\, \boldsymbol{d}(\tau) \qquad \tau=1, 2, ... \tag{13}$$

(iv) if $\lambda_U(\boldsymbol{w})/d\boldsymbol{w} \neq 0$ then set $\tau=\tau+1$ and go to (ii), else return $\boldsymbol{w}(\tau+1)$ as the desired minimum.

Determining the next current point in the iteration process entails two problems. *First*, the search direction $\boldsymbol{d}(\tau)$ has to be determined, that is, what direction in parameter space we want to go in the search for a new current point. *Second*, once the search direction has been found, we have to decide how far to go in the specified direction, that is, step size $\eta(\tau)$ has to be determined.

To solve these problems, normally two types of operation must be carried out: the computation or evaluation of the derivatives of the loss function with respect to the model parameters, and the computation of the parameter $\eta(\tau)$ and the direction vector $\boldsymbol{d}(\tau)$ based upon these derivatives. The evaluation of the loss function is most commonly performed by the backpropagation technique which provides a computationally efficient procedure for doing this. Gradient descent, conjugate gradient and quasi-Newton procedures are characteristically used for the computation of the

parameter $\eta(\tau)$ and the direction vector $\boldsymbol{d}(\tau)$. When $\lambda_U(\boldsymbol{w})/d\boldsymbol{w}$ becomes perpendicular to $\boldsymbol{d}(\tau)$, the algorithm has reached a minimum $\hat{\boldsymbol{w}}_U$. This can be either a global minimum or a suboptimal solution known as local minimum, that is the minimum in a finite neighbourhood. In both cases, very often the solution is not unique, meaning that there exist many permutations of weights and/or hidden units corresponding to the same empirical loss magnitude.

Local search procedures find local minima efficiently and typically work best in unimodal problems. But they have difficulties when the surface of the parameter space is flat [i.e. gradients close to zero], when there is a large range of gradients, and when the surface is very rugged. The search may progress too slowly when the gradient is small, and may overshoot where the gradient is large. When the error surface is rugged, a local search from a random starting point converges to a local minimum close to the initial point and worse solution than the global minimum (Fischer, 2001b).

Global search algorithms employ heuristics to be able to escape from local minima. These algorithms can be classified as probabilistic or deterministic. Of the few deterministic global minimization methods that exist, most apply deterministic heuristics to bring search out of a local minimum. Other methods, like covering methods, recursively partition the search space into subspaces before searching. None of these methods operates well or provides adequate coverage when the search space is large, as is usually the case in neural spatial interaction modelling.

Probabilistic global minimization methods rely on probability to generate decisions. The simplest probabilistic algorithm uses restarts to bring search out of a local minimum when little improvement can be made locally. More advanced methods rely on probability to indicate whether a search should ascend from a local minimum: simulated annealing, for example, when it accepts uphill movements. Other probabilistic algorithms rely on probability to decide which intermediate points to interpolate as new trial parameter vectors: random re-combinations or mutations in evolutionary algorithms (see, for example, Fischer and Leung, 1998).

The success of global search procedures in finding a global minimum of a given function such as λ over $\boldsymbol{w} \in \boldsymbol{W}$ hinges on the balance between an exploration process, a guidance process and a convergence-inducing process. The *exploration process* gives the search a mechanism for sampling a sufficiently diverse set of parameters $\boldsymbol{w}$ in $\boldsymbol{W}$. This exploration process is generally stochastic in nature. The guidance process is an implicit process that evaluates the relative quality of search points and biases the exploration process to move toward regions of high-quality solutions in $\boldsymbol{W}$. The *convergence-inducing process* finally ensures the convergence of the search to find a fixed solution $\hat{\boldsymbol{w}}$. The dynamic interaction among these three processes is responsible for giving the search process its global optimizing character (Hassoun, 1995). An example of a powerful global search procedure is Alopex, a correlation-based method for solving the maximum likelihood problem. The reader interested in details of the procedure is referred to Fischer and Reismann (2002b).

Global search procedures such as Alopex based search – as opposed to local search – have to be used in network training problems where reaching the global optimum is at premium. The price one pays for using global search procedures is increased computational requirements. The intrinsic slowness of such procedures is mainly due to the slow but crucial exploration process. This may motivate the development of a hybrid procedure that uses global search to identify regions of the parameter space containing local minima and gradient information to actually find them (Fischer, 2002a).

5 MODEL ADEQUACY TESTING

The discrimination approach to model selection will identify a particular model $\Omega(\boldsymbol{x}, \hat{\boldsymbol{w}})$ as correctly specified. For a correctly specified model the non-parametric residuals

$$y_u - \Omega(\boldsymbol{x}_u, \hat{\boldsymbol{w}}_U) = e_u \text{ for } u = 1, ..., U \tag{14}$$

are such that $e_u \cong \varepsilon_u = y_u - \boldsymbol{g}(\boldsymbol{x}_u)$. The residuals $\{e_u\}$ can be taken to perform meaningful diagnostic tests about the initial assumptions concerning the stochastic term in the data-generating mechanism [see *Assumption A*]. But because of the non-parametric nature of neural spatial interaction models, satisfying these tests is a necessary, but not sufficient condition for model adequacy. There is always the possibility that a grossly over-parameterized model will satisfy these tests.

Thus, the selected neural spatial interaction estimator is not necessarily a faithful representation of the underlying spatial interaction function $\boldsymbol{g}$ ($\boldsymbol{x}$). There are a number of reasons for this (Zapranis and Refenes 1999):

- *Inadequacies of the estimation procedure*: Convergence issues such as local minima or sensitivity to initial conditions may affect the replicability of the estimation process and distort the relationship between model complexity and estimation error.

- *Incorrect functional form*: In the context of neural spatial interaction models of type (2) this translates to the wrong number H of hidden units; the selected model can be biased and inconsistent, and the variance of the disturbance term incorrectly estimated.

- *Measurement errors in the explanatory and dependent variables*: Omitted observations, approximation errors, outliers etc. can lead to specification bias.

- *Incorrect specification of the error term*: Failure to satisfy the model adequacy tests might be simply due to wrong assumptions about the true nature of the error term, such as *Assumption A*.

Diagnostic checking should be an integral part of model adequacy testing, but can not replace assessing the generalization performance of a model. The standard approach for assessing the generalization performance of a neural spatial interaction model is *data splitting* (see,

for example, Fischer and Reismann, 2002b). This method simulates learning and generalization by partitioning the total data set, say M_U={($\boldsymbol{x}_u$, y_u) with u=1, ..., U}, into three separate subsets: a training [in-sample] set M_{U1}={($\boldsymbol{x}_{u1}$, y_{u1}) with $u1$=1, ..., $U1$}, an internal validation set M_{U2}={($\boldsymbol{x}_{u2}$, y_{u2}) with $u2$=1, ..., $U2$} and a testing [out-of-sample] set M_{U3}={($\boldsymbol{x}_{u3}$, y_{u3}) with $u3$=1, ..., $U3$}. M_{U1} is used for parameter estimation only, while M_{U2} for determining the stopping point before overfitting occurs and to set additional parameters sometimes called hyperparameters. The generalization performance of the model is assessed on the test set M_{U3} using an appropriate performance criterion (such as a normalized mean squared error metric in the context of least squares estimation).

It is common practice to use random splits of the data. The simplicity of this approach is appealing. But randomness enters in two ways: in the splitting of the data samples on the one side and in choices about the parameter initialization of the estimation approach on the other. This leaves one question widely open. What is the variation in generalization performance as one varies training, validation and test sets?

Monte Carlo simulations can provide certain limited information on the behaviour of the test statistics. But the limitation of Monte Carlo simulations is that any results obtained pertain only to the environment in which the simulations are carried out. In particular, the data-generating mechanism has to be specified a priori, and it is often difficult to know whether any given data-generating mechanism is to any degree representative for an empirical setting under study.

To overcome the generally neglected issue of fixed data splitting and its implications Fischer and Reismann (2002a) suggest to combine the purity of splitting the data into three subsets with the power of statistical resampling schemes. The term resampling schemes is used to describe bootstrapping, jackknifing, cross-validation and their variants. These are procedures primarily used for non-parametric estimation of statistical error. They offer a way of obtaining nearly unbiased estimates of model parameters and prediction performance.

In contrast to Monte Carlo simulations bootstrapping and jackknifing do not require a priori specification of the data-generating mechanism,

and can give reasonably accurate approximations of the small-sample distribution properties of $\hat{\boldsymbol{w}}_U$ when *Assumption A* holds. The estimates of bootstrap and cross-validation are asymptotically equivalent. The cross-validation estimates can be viewed as Taylor series approximation of the bootstrap estimates. The main difficulty in applying resampling procedures is that they can be computationally very demanding.

6 BOOTSTRAPPING AND BOOTSTRAP ESTIMATES

Bootstrapping is a computationally intensive non-parametric approach to statistical inference that enables to estimate standard errors by resampling the data in a suitable way (see Efron and Tibshirani, 1993). This idea can be applied to neural spatial interaction modelling in two different ways. One can consider each input-output pattern as a sampling unit, and sample with replacement from the observed input-output pairs in an attempt to take into account the unknown underlying distribution that gave rise to the observations in the first place. This is sometimes called *bootstrapping pairs* since the input-output pairs remain intact, and are resampled as full patterns (Efron and Tibshirani, 1993).

On the other hand, one can treat the model residuals as the sampling units, and create a bootstrap sample by adding residuals to the model fit. This version is termed the *residuals bootstrap*. Bootstrap distribution created in this case is conditional on the actual observations, as opposed to bootstrapping pairs that provides an unconditional bootstrap distribution and may give trustworthy estimates even if the neural spatial interaction model is wrong. This motivates us to briefly consider the pairs bootstrapping rather than the residuals bootstrap approach.

The idea behind the pairs bootstrapping approach is to generate many pseudo-replicates on the training, validation and test sets, then re-estimating the model parameters on each training bootstrap sample, utilizing the associated validation bootstrap sets for stopping the learning process, and testing the out-of-sample performance on the test bootstrap samples. In this bootstrap world, the errors of forecast, and the errors in

the parameter estimates are directly observable (Efron, 1982). The Monte Carlo distribution of such errors can be used to approximate the distribution of unobservable errors in the real parameter estimates and the real forecasts. This approximation is the bootstrap: it gives a measure of the statistical uncertainty in the parameter estimates and the forecasts. The approach will be described for sampling variability, bias and generalization performance estimation.

Generate B independent training bootstrap samples [typically $20<B<200$], by randomly sampling $U1$ times, with replacement from M. Thus

$$M_{U1}^{*b} = \left\{ (\boldsymbol{x}_{u1}^{*b}, y_{u1}^{*b}) \text{ with } u1 = 1,...,U1 \text{ and } b = 1,...,B \right\}. \tag{15}$$

Each bootstrap sample M_{U1}^{*b} is used to compute a parameter vector by minimizing[6]

$$\hat{\boldsymbol{w}}_{U1}^{*b} = \arg\min \left\{ \lambda_{U1}(\boldsymbol{w}^{*b}) \; with \; \boldsymbol{w}^{*b} \in \boldsymbol{W} \text{ and } \boldsymbol{w} \subseteq \Re^p \right\} \tag{16}$$

where p is the number of parameters and $\lambda_{U1}(\boldsymbol{w}^{*b})$ is the empirical loss for the bootstrap sample M_{U1}^{*b}. For a typical neural spatial interaction model this is given by

$$\lambda_{U1}(\boldsymbol{w}^{*b}) = \frac{1}{2\,U1} \sum_{u1=1}^{U1} \left[y_{u1}^{*b} - \Omega\left(\boldsymbol{x}_{u1}^{*b}, \boldsymbol{w}^{*b} \right) \right]^2 \tag{17}$$

where b=1, ..., B. Note that the average bootstrap loss function $\lambda_{U1}(\boldsymbol{w}^{*b})$ converges to $E_{F_{U1}}[\lambda_{U1}(\hat{\boldsymbol{w}}_{U1}^{*b})]$ where the expectation E is taken with respect to the empirical distribution F_{U1} of the bootstrapped samples

[6] Note that it is also necessary to generate B independent validation bootstrap samples denoted as $M_{U2}^{*b} = \left\{ (\boldsymbol{x}_{u2}^{*b}, y_{u2}^{*b}) \text{ with } u2 = 1,..., U2 \text{ and } b = 1,..., B \right\}$ for stopping the estimation process. Furthermore B independent test bootstrap samples denoted as $M_{U3}^{*b} = \left\{ (\boldsymbol{x}_{u3}^{*b}, y_{u3}^{*b}) \text{ with } u3 = 1,..., U3 \text{ and } b = 1,..., B \right\}$ are required for testing the out-of-sample performance of the model.

M_{U1}^{*b}. The observed distribution of the generated $\lambda_{U1}(\hat{\boldsymbol{w}}_{U1}^{*b})$ converges to the distribution of $\lambda_{U1}(\hat{\boldsymbol{w}}_{U1})$ under F_{U1}:

$$E_{F_{U1}}[\lambda(\hat{\boldsymbol{w}}_{U1})] = E_{F_{U1}}[\lambda_{U1}(\hat{\boldsymbol{w}}_{U1})] \tag{18}$$

which can be used as an estimate of the distribution of $\lambda(\hat{\boldsymbol{w}}_{U1})$ under the operating model *F*, i.e. the probability distribution of the underlying function $g(\boldsymbol{x})$.

Standard Error of the Model Output. Because models of form (2) are non-linear the small-sample multivariate distribution of the estimated parameter vector $\hat{\boldsymbol{w}}_{U1}$ is asymptotically normally distributed (White, 1989). But for arbitrarily complex functionals of the parameter vector $\hat{\boldsymbol{w}}_{U1}$, providing estimates for the standard error of the model output can be mathematically intractable. The bootstrapping technique can be used for this purpose.

Let θ be a continuous function of the model parameters, that is $\theta = h(\boldsymbol{w})$, and denote the estimate of θ for the model $\Omega(\boldsymbol{x}, \hat{\boldsymbol{w}}_{U1})$ as $\hat{\theta} = h(\boldsymbol{w}_{U1})$, then the standard error of the estimation as approximated by the sample standard error of the bootstrap replication is

$$\hat{\sigma}_B = \left[\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}^{*b} - \hat{\theta}^{*}(.))^2 \right]^{\frac{1}{2}} \tag{19}$$

with

$$\hat{\theta}^{*}(.) = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}^{*b} = \frac{1}{B} \sum_{b=1}^{B} h(\hat{\boldsymbol{w}}_{U1}^{*b}). \tag{20}$$

The true standard error of $\hat{\theta}^{*} = h(\hat{\boldsymbol{w}}_{U1})$ is a function of the unknown probability density function *F* of θ, that is $\sigma(F)$. With the bootstrap technique one obtains $\hat{F}_{U1}^{*}$ that is supposed to describe closely the empirical probability distribution $\hat{F}_{U1}$, in other words $\hat{\sigma}_B \approx \sigma(\hat{F}_{U1})$. Asymptotically, this means that as the sample size tends to infinity [that

is, $U1 \rightarrow \infty$], the estimate $\hat{\sigma}_B$ tends to $\sigma(F)$. But for finite sample sizes there will be deviations in general.

Bias Estimation. The bootstrap schemes described above can be used to estimate not only the variability of $\hat{\theta}$ but also its bias. Bias can be viewed as a function of the unknown probability density function F of θ, that is $\beta=\beta(F)$. The bootstrap estimate of bias is simply

$$\hat{\beta}_B = \beta(\hat{F}_{U1}) = E^*\left[\theta(\hat{F}^*_{U1}) - \theta(\hat{F}_{U1})\right] \tag{21}$$

where E^* indicates expectation with respect to bootstrap sampling and $\hat{F}^*_{U1}$ is the bootstrap empirical distribution. The bootstrap estimate of bias is

$$\hat{\beta}_B = \tfrac{1}{B}\sum_{b=1}^{B}\left[\hat{\theta}^{*b}(\hat{\boldsymbol{w}}_{UI}) - \hat{\theta}(\hat{\boldsymbol{w}}_{U1})\right]. \tag{22}$$

The bias is removed by subtracting $\hat{\beta}_B$ from the estimated $\hat{\theta}$.

Generalization Performance. For any given neural spatial interaction model, the pairs bootstrap estimate of its generalization performance is given from the following expression:

$$E[\lambda_{U3}(\hat{\boldsymbol{w}}_{U3}, F_{U3})] = (2\ U3\ \ B)^{-1}\sum_{b=1}^{B}\sum_{u3=1}^{U3}[y^{*b}_{u3} - \Omega(\boldsymbol{x}^{*b}_{u3}, \boldsymbol{w}^{*b}_{u3})] \tag{23}$$

where the expectation is taken with respect to F_{U3}, i.e. the empirical distribution of the bootstrapped samples M^{*b}_{U3} instead of the original sample U_3, B is the number of bootstrap samples and $\hat{\boldsymbol{w}}^{*b}$ is the parameter vector estimated for the b-th bootstrapped sample by minimizing Equation (8).

The bootstrap approach is extremely useful in getting a clearer picture of what might be real and what is noise. But the major problem when applied to neural spatial interaction modelling is that the

computational overheads associated with the approach can be quite considerable [see Fischer and Reismann, 2002a].

7 CONCLUDING REMARKS

Neural spatial interaction models are a relatively recent development that can be seen as an example of non-parametric estimation. They are especially attractive in data-rich, but theory-poor spatial interaction contexts. But much of the application development with neural spatial interaction models up to now has been done on an ad hoc basis without due consideration of model adequacy testing in particular. In this contribution we have presented some major principles of a methodology based upon the latest most significant developments in estimation theory, model selection and model adequacy testing theory. It provides the theoretical framework and enables to efficiently utilize neural networks for modelling complex spatial interaction phenomena at any level of spatial resolution.

Much progress has been made in the theory and methodology in recent years. But several important areas remain for further research. The design of a neural network approach suited to deal with the doubly constrained case is still missing. Finding good hybrid optimization procedures for solving the non-convex learning problems is another important issue for further research even though some relevant work can be found in Fischer, Hlavackova-Schindler and Reismann (1999), Fischer and Reismann (2002a, b).

REFERENCES

Alonso, W. (1978). A theory of movement, in: Hansen, N.N. (Ed.), *Human Settlement Systems*. Ballinger, Cambridge [MA], pp. 197-212.

Batten, D.F. and Boyce, D.E. (1986). Spatial interaction, transportation, and interregional commodity flow models, in: Nijkamp, P. (Ed.), *Handbook of*

Regional and Urban Economics, Volume I. North-Holland, Amsterdam, pp. 357-406.

Bergkvist, E. (2000). Forecasting interregional freight flows by gravity models. *Jahrbuch für Regionalwissenschaft,* Vol. 20, 133-148.

Black, W.R. (1995). Spatial interaction modelling using artificial neural Networks. *Journal of Transport Geography,* Vol. 3(3), 159-166.

Carrothers, G.A.P. (1956). An historical review of the gravity and potential concepts of human interaction. *Journal of the American Institute of Planners*, Vol. 22, 94-102.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans*. Society for Industrial and Applied Mathematics, Philadelphia.

Efron, B. and Tibshirani, R. (1983). *An Introduction to the Bootstrap*. Chapman and Hall, New York.

Fischer, M.M. (2002a): Learning in neural spatial interaction models: A statistical perspective. *Journal of Geographical Systems,* Vol. 4(3), 287-299.

Fischer, M.M. (2002b). A novel modular product unit neural network for modelling constrained spatial interaction flows, in: *Proceedings of the IEEE 2002 World Congress on Computational Intelligence: 2002 Congress on Evolutionary Computation*. IEEE Press, Piscataway [NJ], pp. 1215-1220.

Fischer, M.M. (2001a). Spatial analysis in geography, in: Smelser, N.J. and Baltes, P.B. (Eds.): *International Encyclopedia of the Social and Behavioral Sciences*, Vol. 22, pp. 14752-14758. Elsevier, Oxford.

Fischer, M.M. (2001b). Neural spatial interaction models, in: Fischer, M. M. and Leung, Y. (Eds.): *GeoComputational Modelling: Techniques and Applications*. Springer, Berlin, Heidelberg and New York, pp. 195-219.

Fischer, M.M. (2000). Methodological challenges in neural spatial interaction modelling: The issue of model selection, in: Reggiani, A. (Ed.), *Spatial Economic Science: New Frontiers in Theory and Methodology*. Springer, Berlin, Heidelberg and New York, pp. 89-101.

Fischer, M.M. and Getis A. (1999). New advances in spatial interaction theory. *Papers in Regional Science*, Vol. 78, 117-118.

Fischer, M.M. and Gopal, S. (1994). Artificial neural networks: A new approach to modelling interregional telecommunication flows. *Journal of Regional Science,* Vol. 34(4), 503-527.

Fischer, M.M. and Leung Y. (1998). A genetic-algorithm based evolutionary computational neural network for modelling spatial interaction data. *The Annals of Regional Science,* Vol. 32(3), 437-458.

Fischer, M.M. and Reismann M. (2002a). Evaluating neural spatial interaction modelling by bootstrapping. *Networks and Spatial Economics,* Vol. 2(3), 255-268.

Fischer, M.M. and Reismann M. (2002b). A methodology for neural spatial interaction modeling. *Geographical Analysis,* Vol. 34(2), 207-228.

Fischer, M.M., Reismann, M. and Hlavackova-Schindler, K. (2003). Neural network modelling of constrained spatial interaction flows: Design, estimation and performance issues. *Journal of Regional Science,* Vol. 43(1), 35-61.

Fischer, M.M., Hlavackova-Schindler K. and Reismann M. (1999). A global search procedure for parameter estimation in neural spatial interaction modelling. *Papers in Regional Science,* Vol. 78, 119-134.

Fotheringham, A.S. (1983). A new set of spatial interaction models: The theory of competing destinations. *Environment and Planning A,* Vol. 22, 527-549.

Fotheringham, A.S. and O'Kelly, M.E. (1989). *Spatial Interaction Models: Formulations and Applications*. Kluwer, Dordrecht, Boston and London.

Galland, A.R. and White, H. (1988). *A Unified Theory of Estimation and Inference for Nonlinear Dynamic Models*. Basil Blackwell, Oxford.

Hassoun, M.H. (1995). *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge [MA] and London, England.

Isard, W. and Bramhall, D.F. (1960). Gravity, potential, and spatial interaction models, in Isard, W. (Ed.) *Methods of Regional Analysis*. MIT Press, Cambridge [MA], pp. 493-568.

Kullback, S. and Leibler, R.A. (1951). On information and sufficiency. *Annals of Mathematical Statistics,* Vol. 22, 78-86.

Mozolin, M., Thill, J.-C. and Usery, E.L. (2000). Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation. *Transportation Research B,* Vol. 34, 53-73.

Olsson, G. (1965). Distance and Human Interaction. Regional Science Research Institute, Philadelphia, Pennsylvania.

Openshaw, S. (1993). Modelling spatial interaction using a neural net, in: Fischer M.M. and Nijkamp, P. (Eds.), *Geographic Information Systems, Spatial Modeling, and Policy Evaluation.* Springer, Berlin, Heidelberg and New York, pp. 147-164.

Openshaw, S. (1988). Building an automated modelling system to explore a universe of spatial interaction models. *Geographical Analysis,* Vol. 20(1), 31-46.

Reggiani, A. and Tritapepe T. (2000). Neural networks and logit models applied to commuters' mobility in the metropolitan area of Milan, in: Himanen, V., Nijkamp, P. and Reggiani, A. (Eds.), *Neural Networks in Transport Applications*. Ashgate, Aldershot, pp. 111-129.

Sen, A. and Smith, T.E. (1995). *Gravity Models of Spatial Interaction Behavior*. Springer, Berlin, Heidelberg and New York.

Thill, J.-C. and Mozolin, M. (2000). Feedforward neural networks for spatial interaction: Are they trustworthy forecasting tools? in: Reggiani, A (Ed.),

Spatial Economic Science: New Frontiers in Theory and Methodology. Springer, Berlin, Heidelberg and New York, pp. 355-381.

Tobler, W. (1983). An alternative formulation for spatial interaction modelling. *Environment and Planning A,* Vol. 15, 693-703.

Turton, I., Openshaw, S. and Diplock, G. (1997). A genetic programming approach to building new spatial models relevant to GIS, in Kemp, Z. (Ed.), *Innovations in GIS 4*. Taylor & Francis, London, pp. 89-104.

White, H. (1989). Learning in artificial neural networks: A statistical perspective. *Neural Computation*, Vol. 1, 425-464.

Wilson, A.G. (1970). *Entropy in Urban and Regional Planning*. Pion, London.

Wilson, A.G. (1967). A statistical theory of spatial distribution models. *Transportation Research,* Vol. 1, 253-269.

Zapranis, A. and Refenes, A.-P. (1999). *Principles of Neural Identification, Selection and Adequacy. With Applications to Financial Economics*. Springer, London.