Comparison of Regression Methods, Symbolic Induction Methods and Neural Networks in Morbidity Diagnosis and Mortality Prediction in Equine Gastrointestinal Colic

Tuomas Sandholm
sandholm@cs.umass.edu
University of Massachusetts at Amherst
Department of Computer Science
Amherst, MA 01003

Carla Brodley
brodley@ecn.purdue.edu
Purdue University
School of Electrical and Computer Engineering
West Lafayette, IN 47907

Alexandar Vidovic
Hochmoor Animal Clinic
Von Braun Straße 10
48712 Gescher-Hochmoor, Germany

Markus Sandholm
markus.sandholm@helsinki.fi
Helsinki University, Dept of Clinical Sciences
Faculty of Veterinary Medicine
Box 57, FIN-00014 Helsinki, Finland

Abstract

Classifier induction algorithms differ on what inductive hypotheses they can represent, and on how they search their space of hypotheses. No classifier is better than another for all problems: they have selective superiority. This paper empirically compares six classifier induction algorithms on the diagnosis of equine colic and the prediction of its mortality. The classification is based on simultaneously analyzing sixteen features measured from a patient. The relative merits of the algorithms (linear regression, decision trees, nearest neighbor classifiers, the Model Class Selection system, logistic regression (with and without feature selection), and neural nets) are qualitatively discussed, and the generalization accuracies quantitatively analyzed.

1 Introduction

Equine colic—a painful acute abdominal crisis—attributable to gastrointestinal tract disease is the leading cause of death in adult horses. Colic horses require immediate clinical decision making as they often need surgery to open up mechanical obstructions and to remove necrotic parts of the intestine. Endotoxaemia is a typical characteristic of colic. Survival largely depends on host responses. The patients actually die due to a hyperbolic inflammatory response that involves numerous biological pathways. It is not known why some horses (non-survivors) hyperreact and further, which particular regulation mechanism within the inflammatory cascade goes wrong. The process of intestinal colic is dynamic and currently there is no safe indicator to tell the point at which the horse is "over the edge" and cannot be saved. The disease culminates in fluid- and acid-base disturbance, diffuse coagulopathy, multiple organ dysfunctions, and finally death.

Due to the high mortality rate in the surgery and the high cost of the operation (about US$ 10,000), one would like to only operate on horses that A) actually have the disease, and B) will survive the operation. This gives rise to two classification problems: *morbidity diagnosis* (sick or healthy), and *mortality prediction* (survives or dies).

The data consisted of 105 horses with severe gastrointestinal colic; 42 colic horses died within three days and 63 survived the colic episode. Another 52 healthy horses served a a control set in the morbidity diagnosis problem. The predictor data, collected at admission to the clinic, included sixteen features: pulse rate, breath rate and the following laboratory measurements: PCV, HCO_3^-, base excess, anion gap, plasma Na^+, K^+, Cl^-, fibrinogen, D-dimer, endotoxin, the enzymes SDH, GLDH, PLA_2, and a D-dimer to fibrinogen ratio.

Several studies have analyzed the diagnostic and prognostic value of individual features and feature combinations in equine colic; for a review, see Sandholm et al. (1995). High pulse rate associated with high packed cell volume, dull color of mucus membranes, delayed oral mucus capillary refill time, disturbances in acid-base parameters—such as increased lactate or anion gap—and a hypercoagulative condition have been used as predictors for poor prognosis. In other words, pathophysiological knowledge has guided decision making. Multiple logistic regression has been used to combine various predictors for most accurate prediction so far (Reevers et al. 1992). Recently Sandholm et al. (1995) reported that increasing heart rate and plasma D-dimer together with decreasing chloride was a typical risk factor for non-survival, and that these three features could be used to enhance the accuracy of the logistic regression.

This paper discusses the application of symbolic induction algorithms, neural networks, and statistical techniques to morbidity diagnosis and mortality prediction in equine colic. There were three objectives to this research. The first was to find the method that results in the most accurate classification of morbidity and mortality by intelligently using different measured features of a patient simultaneously. The second was to gain further insight into the strengths and weaknesses of the available classifier construction algorithms. The third objective was to determine which features are actually useful in the prediction and should therefore be measured from horses in clinics.

The remainder of this paper is organized as follows. Section 2 describes how the the different classifiers were evaluated. Section 3 discusses the different classifier induction methods, and presents qualitative comparisons and quantitative evaluation results. Section 4 discusses the pruning of features that are not relevant. Section 5 concludes and presents directions for future work.

2 Experimental classifier evaluation

To allow for fair comparison, each of the various classifier construction methods was applied using the same experimental conditions. To assess the ability of each

method to produce an accurate classifier we average, for each method, the results of ten runs. For each run we split the original data randomly into two sets; 90% of the data was used to form the classifier and the remaining 10% was used to evaluate the classifier. We hold back 10% of the data for testing because the goal of a classifier construction method is to create a classifier that will provide a high degree of accuracy when used to classify previously unseen cases.[1] For each of the 10 splits, the few missing feature values were replaced with the class average observed for the feature in the training set.

To ensure that the distribution of cases across the classes of sick and healthy (similarly died and survived) is the same in the training and test sets, we first sorted the data into these two groups. We then dealt the horse cases out randomly to the training and test sets in the specified proportions (90 and 10). Each method was run using the same partitions. In the experiments we report the average of each method's *generalization accuracy*: accuracy on the independent test sets.

3 Classifier induction methods

In addition to traditional classification methods such as linear regression and logistic regression, several dozen classifier construction algorithms have been developed in the last few decades in the machine learning community, including various versions of perceptron (Nilsson 1965), version space (Mitchell 1977), decision tree (Quinlan 1986), instance-based (Duda & Hart 1973), and neural net algorithms (Rumelhart & McClelland 1986). The results of empirical comparisons of existing algorithms illustrate that each algorithm has a selective superiority: it is best for some but not all classification tasks (Brodley 1993). Selective superiority arises because each learning algorithm searches within a restricted hypothesis space defined by its class of models. For example, the class of first-order linear regression models is not appropriate when the data is best fit by a second-order linear regression model. In addition, each method has a specific strategy for exploring its hypothesis space; exploring the entire is space is typically computationally infeasible.

The existence of selective superiority can also be easily shown by a theoretical argument. Say that one wants to show that classifier A is better than classifier B on all classification problems in terms of accuracy on feature vectors that are not in the training set. A classifier is a mapping from feature vectors to classes. For classifier A to be better than B, these two classifiers have to have different classifications for some feature vectors. If both classifiers are consistent with the training set, then the feature vectors on which the two classifiers predict different classes cannot be the training set. Let an adversary pick the correct class for these feature vectors. Now, the adversary can pick so that A misclassifies all of them, while B classifies all of them correctly. Thus for this labeling, B is a better classifier than A, which disproves the attempted argument. Thus no classifier can be better than another in general in the sense of generalization accuracy, because an adversary can refute this claim.

This paper compares six methods for constructing classifiers in the morbidity diagnosis and mortality prediction problems: linear regression, decision trees, nearest neighbor classifiers, the Model Class Selection system, logistic regression, and 3-layer feedforward neural networks. These methods, their relative merits, and the results regarding classification accuracy are discussed in the following subsections.

3.1 Linear regression

A *linear threshold unit* (LTU) (Nilsson 1965) is a binary test of the form $W^T Y \geq 0$, where Y is a vector consisting of a constant 1 and the n features that describe the instance. W is a vector of $n+1$ coefficients, also known as weights. If $W^T Y \geq 0$, then the LTU infers that Y belongs to one class A, otherwise the LTU infers that Y belongs to the other class B.

To find the set of weights that leads to an accurate classifier, we used the Recursive Least Squares (RLS) Procedure (Young 1984). RLS, invented by Gauss, is a recursive version of the Least Squares (LS) Algorithm. An LS procedure minimizes the mean squared error, $\sum_i (y_i - \hat{y}_i)^2$ of the training data, where y_i is the true value and $\hat{y}_i$ is the estimated value of the dependent variable, y, for feature vector i. For discrete classification problems, the true value of the dependent variable (the class) is either c or $-c$. In our implementation of the RLS procedure we use $c = 1$. Note that a procedure that minimizes the mean squared error between the estimates and the true value of the dependent variable is a maximum likelihood estimator for the coefficients. However, although RLS is a MLE, if the data are not linearly separable then the LTU will not be able to capture the exact underlying structure of the data.

3.2 Decision tree

A *univariate decision tree* is either a leaf node containing a classification or a node containing an attribute test. In the latter case, the node contains a branch to a decision tree for each value of the attribute. To classify a feature vector using a decision tree, one starts at the root node and finds the branch corresponding to the value of the test attribute observed in the feature vector. This process repeats at the subtree rooted at that branch until a leaf node is reached. The feature vector is then assigned the class label of the leaf. One well-known approach to constructing a decision tree is to grow a tree until each of the terminal nodes (leaves) contains training instances from a single class only. The tree can then be pruned back with the objective of reducing the misclassification rate. Our algorithm uses reduced error pruning (Quinlan 1987), which replaces a subtree with a leaf if it reduces the error on a set of data independent from the training data. (Note that this requires that we retain a portion of the training data to use for pruning the tree).

To select a test for a node in the tree, we choose the test that maximizes the information-gain ratio metric (Quinlan 1986). Univariate decision tree algorithms require that each test have a discrete number of outcomes. To meet this requirement, each ordered feature A_i is mapped to a set of unordered features by finding a set of Boolean tests of the form $A_i > b$, where b is

[1] If the evaluation were done on the same data as the training, some methods would achieve 100% accuracy, because they would remember the classes of the training examples correctly.

in the observed range of A_i. Our algorithm finds the value of b that maximizes the information-gain ratio. To this end, the observed values for A_i are sorted, and the midpoints between class boundaries are evaluated (Quinlan 1986; Fayyad & Irani 1992).

Decision trees are restricted to placing boundaries in the feature vector space that are orthogonal to each of the feature axes. Therefore if there is any relationship among the features it may not be captured well. On the other hand, unlike linear machines, decision trees are not restricted to dividing the feature vectors linearly into classes, because any section of the feature vector space that is separated from other parts of the space by a boundary, can be further split into subspaces that carry different class labels. Decision trees are perhaps the most human-understandable learning method, which is important for trying to explain classification decisions.

3.3 Nearest neighbor classifier

A *k-nearest neighbor classifier* (Duda & Hart 1973) is a set of n instances, each from one of m classes, that are used to classify feature vectors according to the majority classification of the feature vector's k nearest neighbors. In this version of the algorithm each instance in the training data presented to the algorithm is stored.[2] To determine how near a feature vector is to another, the Euclidean distance between the two is computed. In our experiments k was set to one.

Nearest neighbor classifiers have a less restrictive hypothesis space than linear discriminants and decision trees; they form piece-wise linear boundaries in the feature vector space. However, if some of the features that describe the data are irrelevant or noisy then a nearest neighbor classifier may be inaccurate. One solution to this problem is to use a learning method to define weights for each for the features (Aha 1992; Cost & Salzberg 1993). Indeed, in Section 4 we illustrate that only a subset of the features are relevant in the diagnosis and prediction problems in equine colic.

3.4 Model Class Selection (MCS) system

Given a data set, it is often not clear beforehand which algorithm will yield the best performance. In such situations, someone wanting to find a classifier for the data will be confused by the myriad of choices, and will need to try many of them in order to be convinced that a better classifier will not be found easily. Recently, the *Model Class Selection (MCS) system* has been developed to overcome this problem. MCS applies knowledge about the biases (restricted hypothesis spaces and limited ways of exploring those spaces) of linear discriminant functions, decision trees, and nearest neighbor classifiers to conduct a recursive automatic algorithm search.

MCS uses characteristics of the given data set, in the form of feedback from the learning process, to guide a search for a tree-structured hybrid classifier. Heuristic knowledge about the data characteristics that indicate that one algorithm is better than another is encoded in a rule base. The approach does not assume that the entire data set is best learned using a single algorithm; for some data sets choosing to form a hybrid classifier will produce a more accurate classifier, and MCS attempts to determine these cases. The results of an empirical evaluation illustrate that MCS achieves classification accuracies equal to or higher than the best of its primitive learning components for each of a variety of data sets, demonstrating that the heuristic rules effectively select an appropriate algorithm(s). Details of these experimental results and of the MCS system can be found in Brodley (1995).

Table 1 shows the generalization accuracy of MCS and its component learning algorithms. For the mortality data set, MCS has higher accuracy than its primitive algorithms. For the morbidity data set, every method except for decision trees performs equally well.

3.5 Logistic regression

Logistic regression is a well-known statistical method for building classifiers. The idea is to use the *logit* transformation $ln(c/(1-c))$ to recode the classification c which is between zero and one. Then a linear model is used to predict $ln(c/(1-c))$ based on the input features. The maximum likelihood estimator is acquired via an iterative least squares method.

Again, for each split separately, the classifier was constructed based on the training set and evaluated on the test set. Before each regression, *collinearity* was removed. If a feature was highly correlated with a linear combination of other features, that feature was dropped from the model. This was repeated with the remaining features until all such collinearities were removed.

In two-class classification problems, one class is associated with the values of c close to zero, and the other with values close to one. The classification threshold need not be at $c = 0.5$. It was chosen so as to maximize classification accuracy on the training data.

Logistic regression has a very restricted model class: like linear regression, it can only divide the feature vector space into two regions—one for each class—using a hyperplane. Yet, it has advantages over linear regression. First, it never associates a feature vector with a class value that is out of range, i.e. greater than one or less than zero. Second, it tends to assign class values close to one or zero unlike linear regression, which linearly assigns values in between also.

Table 1 shows the generalization accuracy of MCS, its component algorithms, and logistic regression. For the mortality data set, logistic regression did worse than a linear discriminant and MCS. For the morbidity data set, it outperformed the other methods.

Method	Mortality	Morbidity
Linear discriminant function	66.0	95.3
Decision tree	62.0	94.7
Nearest neighbor classifier	64.0	95.3
Model Class Selection system	68.0	95.3
Logistic regression	65.0	98.8

Table 1: *Average generalization accuracy (%).*

3.6 Neural net

We also examined how well the classification problems can be solved using artificial neural nets. Unlike the other methods, the neural network is not a single

[2] This is distinct from the entire set of training data; the filtering mechanism may determine that only part of the data should be given to the k-nearest neighbor classifier.

method but a collection. To instantiate a specific net, one needs to decide the topology—e.g. number of hidden units and connections—and the parameters for the learning algorithm that updates the weights in the net. In the experiments, each input feature is an input to the net, resulting in sixteen input units. The inputs were not coded or normalized in any way. We used a three-layer feedforward neural net architecture, because it can represent any mapping from inputs (from a closed and bounded part of the feature vector space) to outputs, i.e. it has no restrictions on the model class (Hecht-Nielsen 1991). Each input unit was connected to each hidden unit, and each hidden unit was connected to the single output unit. We denoted one class with an output of 1 and the other class with a 0. During testing, we used a classification threshold of 0.5 on the output of the net. The input units simply output their input. The hidden units and output unit output according to the logistic function (Rumelhart & McClelland 1986). The weights of the connections were updated using the standard backpropagation rule (Rumelhart & McClelland 1986). Backprop has two parameters: *learning rate* determines how fast the weights in the net are adjusted and *momentum* determines how slow it is to change the weight changes themselves on each update (Rumelhart & McClelland 1986). In our experiments, learning rate was varied and momentum was set to one tenth of the learning rate.

We experimented with different net topologies by varying the number of hidden units from a low of three to a high of 31. By exploratory data analysis we narrowed the number of hidden units for the tests to five, ten and twenty. The results, which are very sensitive to these changes in topology, are shown in Table 2. From the results it is clear that five hidden units was too few. On the other hand, twenty seems to be unnecessarily many on the mortality task, but is a good number for the morbidity problem. Increasing the number of hidden units increases the net's degrees of freedom—and therefore also the representation power[3]—and usually provides better accuracy on the training data, but may result in lower accuracy on previously unseen test data due to overfitting of the training data.

Each training session included 10,000 passes (*epochs*) through the training data. After each epoch, the classification accuracy on the test data was measured. When using neural nets in practise, it is difficult to know when to stop training. With too few epochs, the net will not have enough time to learn. With too many epochs, the net usually overfits the training data, causing a decrease in classification accuracy on the test data. On the equine data sets the optimal point to stop training varied between net topologies and learning algorithm parameterizations. Even more problematically, it varied widely between different splits of the data for a given topology and parameterization.

Each entry of Table 2 reports four different results. The first number reports the average of the highest observed classification accuracy for each test set, i.e. when the net had already learned, but when it had not yet overfit the training data. This number was measured at the best number of training epochs for each of the ten splits separately.[4] According to these numbers, the neural net outperforms the other methods—particularly on the difficult mortality prediction problem. But this is an unfair comparison because the net uses the test data in choosing the classifier: it generates a different classifier at each training epoch (based on the training set), and the best classifier is chosen (based on the test set) over all epochs. In practice, one would not have this information unless part of the training data was retained for this task, which in turn could result in lower accuracy because the net would be trained using fewer training instances. As a more traditional comparison, the generalization accuracies were also analyzed at fixed numbers of training epochs (100, 1000, and 10000). This degraded generalization significantly, which can be seen in Table 2.

Hidden units	Mortality	
	Learning rate 0.01	Learning rate 0.001
5	**62.0**, 59.0, 59.0, 60.0	**63.0**, 57.0, 57.0, 53.0
10	**72.0**, 56.0, 62.0, 60.0	**70.0**, 63.0, 59.0, 53.0
20	**71.0**, 58.0, 59.0, 61.0	**70.0**, 56.0, 56.0, 57.0
Hidden units	Morbidity	
	Learning rate 0.01	Learning rate 0.001
5	**75.3**, 66.7, 66.7, 66.7	**91.3**, 66.7, 80.0, 78.0
10	**86.7**, 71.3, 66.7, 68.0	**99.3**, 66.7, 88.0, 86.7
20	**92.7**, 68.0, 66.7, 66.7	**99.3**, 77.3, 94.7, 98.0

Table 2: *Average generalization accuracy (%). The first number is the accuracy when training is stopped on the best epoch for each of the ten training sets separately. "Best" is measured as classification accuracy on the test set. The second number is the accuracy after 100 epochs, the third for 1000, and the 4th for 10000.*

4 Feature selection

In the method comparison experiments above, all sixteen features were used. It is sometimes advantageous to lower the dimensionality of the feature vector space by ignoring some features. This allows a finite set of training instances to populate the space more densely, but may ignore significant predictors.

In our feature selection experiments, both the training data and test data were used together. To begin with, collinear features were removed as in Section 3.5. Then feature selection was performed exhaustively by running a linear regression on each possible combination of the features. The criterion for the goodness of the model was based on the adjusted R^2 statistic, which takes into account both the residual sum of squares, and the number of features in the model (Statistix User's Manual 1992). In general, the model with the higher adjusted R^2 was preferred, but when the difference in the adjusted R^2 was small for two models (less than 0.0225), the model with fewer features was chosen. The best model for the morbidity problem contained four features: endotoxin, K^+, pulse rate, and D-dimer. The best model for the mortality problem contained three features: Cl^-, D-dimer and

[3] With n inputs, $2n+1$ hidden units suffice to represent any mapping from inputs to outputs (Hecht-Nielsen 1991).

[4] In 92% of all the splits of the mortality data, highest generalization accuracy was achieved by 500 epochs. Similarly, in 80% of the splits of the morbidity data, highest accuracy was achieved by 500 epochs.

pulse rate.[5]

Next we analyzed the accuracy of logistic regression using these reduced feature sets. For each split we trained the model on the training data and tested it on the separate test data. The average accuracy on the mortality problem increased to 73% but on the morbidity problem it dropped to 95.6%. These numbers are not directly comparable to those in Table 1 because the test data was used for feature selection as described above—and thus implicitly for classifier construction. When trained and evaluated on the same data (training and test data combined), the classification accuracy of logistic regression on the mortality problem increased to 77.5%.

5 Conclusions and future research

Classifier induction algorithms differ on what inductive hypotheses they can represent, and on how they search their space of hypotheses. For example, linear and logistic regression have very restricted hypothesis spaces while three-layer neural nets have an unrestricted hypotheses space. Yet, no classifier is better than another for all problems: they have selective superiority. In this paper we empirically compared six classifier induction methods in the domains of diagnosing equine colic and predicting its mortality.

Morbidity diagnosis was easy for all methods. The average generalization accuracy varied between 94.7% and 99.3%. Logistic regression and neural nets had the highest accuracies, but the differences between the methods were small. High accuracy was achievable because endotoxin in plasma is an accurate discriminator between sick patients and controls. *Mortality prediction* was difficult for all methods. The average generalization accuracy varied between 62.0% and 72.0%. Neural nets and MCS had the highest accuracies. For both classification tasks, MCS had higher accuracy than any of its base-level methods. The neural net results are not directly comparable to the other methods because test data was used in choosing the number of hidden units, the learning rate, and the best time to stop training and generalization accuracy is sensitive to these choices.

Decreasing the number of features reduced the generalization accuracy of logistic regression in morbidity classification, but enhanced it in mortality prediction from 65% to 73%. This is the best generalization accuracy achieved on the problem. Test data was used in feature selection, but not in running the logistic regressions. The best classifier for mortality prediction contained only three features.

The classifiers provide a convenient way of performing rapid "horse-side" prediction based on a large set of relatively easily measurable patient features. Future work would include tailoring classifiers to individual horse clinics based on their previous cases. Comparison of the observed mortality with the predicted mortality would allow a clinic to monitor how well it—or an individual surgeon—is performing. Classifiers trained on case data from other clinics would also allow comparisons across clinics. When the therapy (surgery) failures are analyzed against the predicted nonsurvivals, one can minimize the effect of the status of the horse and extract the effect of therapy. Therefore, the classifiers would allow therapy success to be analyzed even if the status of the horses varies from patient to patient and from clinic to clinic. Continuous updating with new cases would indicate the performance trends of the clinic and of each surgeon.

Another extension would be to analyze the ongoing change in a patient's features. With current methodology, it is quite difficult to analyze the patient's changing condition during the short disease process and to draw conclusions. Apparently the importance of individual features changes at different stages of the disease. Here, rapid classifier-based decision making could certainly help.

[5] Feature selection methods using logistic regressions with *forward addition* and *backward elimination* also found the same feature combinations to be the most relevant ones. This happened even though instances with missing feature values were ignored and the sixteenth feature (a ratio of two other features) was not included among the alternatives (Sandholm et al. 1995).

References

Aha, D. W. 1992. Tolerating Noisy, Irrelevant, and Novel Attributes in Instance-Based Learning Algorithms. *International Journal of Man-Machine Studies* 36:267–287.

Brodley, C. E. 1993. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Machine Learning: Proceedings of the Tenth International Conference*, 17–24, Morgan Kaufmann.

Brodley, C. E. 1995. Recursive automatic bias selection for classifier construction. *Machine Learning* 20: 63–94, Kluwer, Hingham, MA.

Cost, S. and Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning* 10:57–78.

Duda, R. O. and Hart, P. E. 1973. *Pattern classification and scene analysis.* Wiley & Sons, New York.

Fayyad, U. M. and Irani, K. B. 1992. The attribute selection problem in decision tree generation. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 104–110, MIT Press.

Hecht-Nielsen, R. 1991. *Neurocomputing.* Addison-Wesley, Reading, MA.

Mitchell, T. M. 1977. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 305–310, Morgan Kaufmann.

Nilsson, N. J. 1965. *Learning machines.* McGraw-Hill, New York.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learning* 1: 81–106, Kluwer, Hingham, MA.

Quinlan, J. R. 1987. Simplifying decision trees. *International Journal of Man-Machine Studies* 27: 221–234.

Reevers, M. J., Curtis, C. R., Salman, M. D., Stashak, T. S. and Reif, J. F. 1992. Validation of logistic regression models used in the assessment of prognosis and the need for surgery in equine colic. *Prev. Vet. Med.* 13: 155–172.

Rumelhart, D. E. and McClelland, J. L. 1986. *Parallel distributed processing.* MIT Press, Cambridge, MA.

Sandholm, M., Vidovic, A., Puotunen-Reinert, A., Sankari, S., Nyholm, K. and Rita, H. 1995. D-dimer improves the prognostic value of combined clinical and laboratory data in equine gastrointestinal colic. *Acta vet. scand.* 36, 2: 255–272.

Statistix User's Manual. 1992. Version 4.0. (c) 92 Analytical Software.

Young, P. 1984. *Recursive estimation and time-series analysis.* Springer-Verlag, New York.