Prediction of ungulates abundance through local linear algorithms

Mauro Bianchi, Giorgio Corani [a,*], Giorgio Guariso [a], Ciro Pinto

[a] *Dipartimento di Elettronica ed Informazione*
Politecnico di Milano
Via Ponzio 34/5
20133 Milano - Italy

PREPRINT TO APPEAR ON:
Environmental Modelling & Software

Abstract

We use a local learning algorithm to predict the abundance of the Alpine ibex population living in the Gran Paradiso National Park, Northern Italy. Population abundance, recorded for a period of 40 years, have been recently analyzed by Jacobson et al. (2004), who showed that the rate of increase of the population depends both on its density and snow depth. In the same paper, a threshold linear model is proposed for predicting the population abundance.

In this paper, we identify a similar linear model in a local way, using a lazy learning algorithm. The advantages of the local model over the traditional global model are: improved forecast accuracy, easier understanding of the role and behaviour of the parameters, effortless way to keep the model up-to-date.

Both data and software used in this work are of public domain; therefore, experiments can be easily replicated and further discussions are welcome.

Key words: lazy learning, population dynamics, Alpine Ibex, time series analysis, nonparametric regression.

* Corresponding Author:
e-mail: corani@elet.polimi.it

Software availability

Name of software: Lazy Learning Toolbox for use with Matlab
Website: http://iridia.ulb.ac.be/˜lazy/
Developer: Mauro Birattari and Gianluca Bontempi.
Affiliation: IRIDIA - Université Libre de Bruxelles - Brussels, Belgium
Year first available: 1999
Software required: Matlab©(www.mathworks.com) and a C compiler
Program language: The "Lazy Learning Toolbox for use with Matlab" consists of four functions, written in C language for computational efficiency. They are designed to be compiled and subsequantly invoked from a matlab shell.
Availability and cost: Open-source software, publicly available from the website.
Further notes: A more recent implementation of Lazy Learning, realized by the same authors, is provided for R, an open source language for data analysis and graphics. The lazy package for R is available from http://cran.r-project.org/src/contrib/Descriptions/lazy.html.

1 Introduction

We study the population of Alpine ibex (*Capra ibex*, Fig. 1), a medium size ungulate living in the Gran Paradiso National Park (Northern Italy). The Park sizes about 703 km^2; hunting is not allowed inside or close to the park, and large predators have been absent over the past 100 years. According to the literature overview presented in Jacobson et al. (2004), if large predators are rare or absent, the changes in ungulate populations can be modelled by considering climate forcing and density-dependence only. Censuses of the population, together with meteorological observations (temperature, precipitation and snow depth) were collected over the period (1960-2000), and the dataset is publicly available from the Ecological Archives website. [1] The plots of the population time series (Fig. 2) shows that the population fluctuated in the first period (1960-1980) between 2800 and 4000 individuals. In the period 1982-1990, the population grew rapidly up to 5000 individuals to then decrease again under 4000. The influence of snow depth on the population abundance is noteworthy: the most important snow peaks resulted in important decreases of the population, while the population increase between 1982 and 1993 corresponds to a period of quite low snow. Indeed, the minimum snow depth occurred in 1991 with a value of only 36 cm, in comparison to a mean value around 1.5 m in the first 20 years of the series.

[1] www.esapubs.org/archive/ecol/E085/043/appendix-A.htm

In this paper, we propose a predictive model of the population, considering the effects of both snow depth and population density. Predictions are computed via a local linear algorithm known as *lazy learning* (Birattari et al., 1999a).

The remaining of the paper is organized as follows: the next section briefly introduces the threshold model developed by Jacobson et al. (2004); section 3 presents the local linear algorithm, section 4 compares the simulation results of the threshold and the local model. Section 5 presents the conclusions.

2 A threshold population model

The ibex population dynamics has been recently analyzed in Jacobson et al. (2004). On the basis of several statistical tests, the dependence of the rate of increase of the population on both the population size (density-dependence) and the snow depth has been rigorously detected. In the same paper, the following demographic model was selected as the best compromise between precision and robustness from 20 candidates structures:

$$Y_{t+1} = ln\left(\frac{N_{t+1}}{N_t}\right) = \begin{cases} a + b_1 S_t + c_1 N_t S_t \text{ if } \; S_t < \overline{S} \\ a + b_2 S_t + c_2 N_t S_t \text{ if } \; S_t > \overline{S} \end{cases} \tag{1}$$

where N_t denotes the population size in year t, S_t the average snow depth in the same year and $N_t S_t$ the interaction of these two quantities. In practice, the model is constituted by two linear regressions with the same arguments. Note that the estimate of a is unique, whereas parameters b and c are different in the two snow conditions; they are fitted either using data referring to years with high snow levels ($S_t > \overline{S}$) or below the snow depth threshold ($S_t < \overline{S}$). The threshold value $\overline{S}$ is set by the Authors as equal to the mean plus one half the standard deviation of the snow ($\overline{S}$=154 cm).

In this paper, we identify the same type of regression in a local way, using lazy learning.

3 The local linear prediction algorithm

According to Fox (2000), local linear modelling can be seen as a case of nonparametric regression; the main feature of the methods belonging to the family of nonparametric regression is that the approximating input-output function is not specified in advance.

The essential idea in local modelling is to renounce to a global description of the system under study, focusing on its linear approximation only in the neighborhood of a condition of interest (*query-point*). See for instance the one-dimensional sample of Fig. 3.

In particular, the lazy learning algorithm proposed by Birattari et al. (1999a) is a local linear approach that defers the processing of the dataset until a request for prediction is received; when this happens, an identification procedure takes place and an *ad hoc* local model is designed. In our case, query-points are constituted by vectors of type $[1, S_t, N_t S_t]$, i.e. the variables appearing in the right-hand side of Eq. 1, in correspondence of which a prediction is required; coefficients a, b, c which map a given query-point on the expected rate of increase Y_{t+1} are locally estimated for each prediction.

As the algorithm receives a request of prediction, it evaluates the distance between the current query-point and the available historical samples; to avoid issues with the measurement units of the different variables, it is necessary to compute such distance on normalized data. Hence, we standardize the variables to have mean 0, and standard deviation 1 on the training data; then, we compute the distance between the query-point q and an historical samples x using the Manhattan metric $D(q, x) = |S_{\ q} - S_x| + |N_q S_q - N_x S_x|$. Historical samples are then ranked according to their distance from the current query-point.

Let us denote as k the number of neighbors used for the identification of a local model. The parameters of the linear model are first estimated using a minimum number of neighbors k_{min} (i.e., the k_{min} historical samples closest to the current query-point). Then, k is increased of one unit, and further estimation is performed using the $(k_{min} + 1)$ closest samples. Different local models are estimated this way, up to a maximum k_{max}. Typical values of k_{min} and k_{max} are for instance $2d$ and $5d$, where d represents the number of free parameters of the model.

Deciding the optimal number of neighbors to be used for the identification of the local model is a critical issue; a too small k leads to an excessive variance of the parameter estimates, thus worsening the accuracy of the model; on the other hand, a large k can lead to a poor representation of non-linearities, since the model tends to a global linear model. An automatic criterion, suitable to tune the number of neighbors on a query-by-query basis, is for instance leave-one-out cross validation (Birattari et al., 1999a): the dataset corresponding to the current value k is split into a training subset of cardinality $(k - 1)$ and a cross-validation subset, containing just the remaining sample. Parameters are estimated on the training set, while the error is measured on the cross-validation sample (*leave-one-out error*); by iterating the procedure, a vector of k leave-one-out errors is generated. The model showing the lowest local average leave-one-out error is finally chosen.

Since the distances from the query-point are evaluated all over again for each prediction, the entire identification dataset is always kept in memory. Lazy learning is hence a *memory-based* approach. On the one hand, this is a drawback in terms of computational requirements; on the other hand, this means that lazy learning can be effortless kept up-to-date by simply adding the new samples to the set of historical data in memory. Such a feature is not very important in the case at hand, since every year only one new sample is added, but could be very welcome when the dynamics of the environmental phenomenon is faster, such as for instance in hydrology or air pollution monitoring, when samples are collected at much higher frequency, and a fast update procedure is a great advantage.

The lazy learning package has been developed by IRIDIA and Machine Learning Group of University of Bruxelles, and is available as open source software, as detailed at the beginning of the paper. A noteworthy feature of such a software is the use of recursive techniques for both model identification and leave-one-out cross-validation, which greatly speeds up the algorithm (Birattari et al., 1999b).

Lazy learning performances in regression and in time series prediction have been shown to be better or as good as a number of state-of-the-art machine learning algorithms (Bontempi, 1999). It has been moreover applied with good results in industrial electronics (Villacci et al., 2005). The only application in the environmental field is, to the best of our knowledge, the prediction of PM_{10} and O_3 in Milan (Corani, 2005). Lazy learning showed almost an identical forecast accuracy to neural networks, but much faster design and update times, and a much better possibility of analyzing the role of model parameters.

4 Simulation results

We compare the forecast accuracy of the threshold linear model in Eq. 1 and of the local learning algorithm, that hence corresponds to a whole set of regressions with the same structure, each identified with the set of the closest data points. We use the first 20 years of data as training set, and the remaining part of the dataset for validation. Figure 4 shows the 1-year ahead predictions computed by the two approaches on the validation data; table 1 reports the mean absolute error (MAE) and the ratio of the mean absolute error to the mean of the population abundance (error ratio) on validation data. Both methods actually return the expected rate of increase $\hat{Y}_{t+1}$, from which the predicted abundance can be easily computed as $\hat{N}_{t+1} = N_t exp\left(\hat{Y}_{t+1}\right)$.

This helps explaining the apparently odd behaviour in years 1990-92. In 1990, a relative high snow depth occurs at a high population value. Since the models forecast the relative population variation $\hat{Y}_{t+1}$, and not its abundance $\hat{N}_{t+1}$, the absolute error turns out to be larger when N_t is high. In addition, a similar situation never occurred in the years of the training set.

The local model shows lower MAEs and error ratio than the threshold model.

Figure 5 shows the estimates of parameter c versus the snow depth values S_t contained in the different query-points. Although both approaches show a decrease of c with the snow depth, the threshold model is characterized by a sharp switch of the parameters around $\overline{S}$; on the contrary, the estimates of the local linear learning have a smooth behavior, that contributes to the improvement of the forecast accuracy and is probably more acceptable from an ecological point of view. In fact, as already noted by Jacobson et al. (2004), the long life expectancy of ibex (about 15 years) tends to filter out rapid variations of the environmental conditions.

5 Conclusions

The local linear algorithm used in this work proved to be suitable for the prediction of the time series of the considered population. It is effective in modelling non-linearities thanks to locality, coupled with a very fast design, due to the use of computationally efficient linear algorithms.

We use this lazy learning procedure to locally identify the demography of Alpine ibex living in Gran Paradiso National Park. Local identification avoids the abrupt change of model parameters in correspondence of the threshold, that is the main drawback of threshold approaches, and improves the forecast accuracy. However, its most interesting feature from an ecological point of view is probably the fact that, for each query point, it is possible to assess in a straightforward way the sensitivity of the local models to the input variables. Such a feature is typical of local modelling approaches and may also help in the determination of a functional relation between the parameters and the input values. Though these results are obtained here in a specific context, we think that lazy learning constitutes an appealing tool for the analysis of other environmental datasets; its diffusion in future studies should also be favoured by the availability of the toolboxes as open source software.

References

Birattari, M., Bontempi, G., Bersini, H., 1999a. Lazy learning for modeling and control design. Int. J. of Control 72, 643–658.

Birattari, M., Bontempi, G., Bersini, H., 1999b. Lazy learning meets the recursive least-squares algorithm. In: M.S. Kearns, S.A. Solla, D. C. (Ed.), Advances in Neural Information Processing System. pp. 375–381.

Bontempi, G., 1999. Local learning techniques for modeling, prediction and control. Ph.D. thesis, Université Libre de Bruxelles, Belgium.

Corani, G., 2005. Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning. Ecological Modelling 185, 513–529.

Fox, J., 2000. Multiple and Generalized Nonparametric Regression. Sage, Thousand Oaks CA.

Jacobson, A., Provenzale, A., Von Hardenberg, A., Bassano, B., Festa-Bianchet, M., 2004. Climate forcing and density dependence in a mountain ungulate population. Ecology 85, 1598–1610.

Villacci, D., Bontempi, G., Vaccaro, A., Birattari, M., 2005. The role of learning methods in the dynamic assessment of power components loading capability. IEEE Transactions on Ind. Electr., 52, 280 – 290.

Tables

	threshold	*local model*
MAE	315	297
error ratio	.078	.073

Table 1. Error statistics on validation data, with reference to the predicted abundances $\hat{N}_t$.

Figure Captions

Fig. 1. Alpine ibex (*Capra ibex*)

Fig. 2. Alpine ibex counts and winter snow depth over the period 1960-2000.

Fig. 3. Local linear model. The linear model estimates the value of the function in q by fitting the six nearest neighbors (bigger dots).

Fig. 4. Out-of-sample predictions computed by the threshold and the local model

Fig. 5. Estimates of parameter c as a function of snow depth. Empty circles represent training data, while solid circles refer to testing values.

Figures

Fig. 1. Alpine ibex (*Capra ibex*)

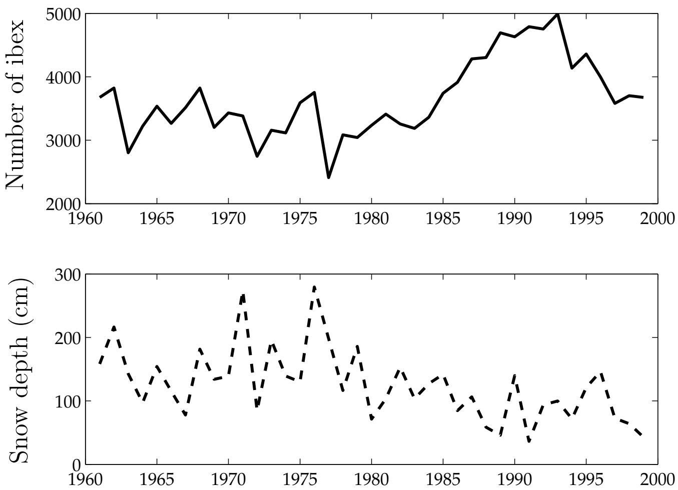

Fig. 2. Alpine ibex counts and winter snow depth over the period 1960-2000.

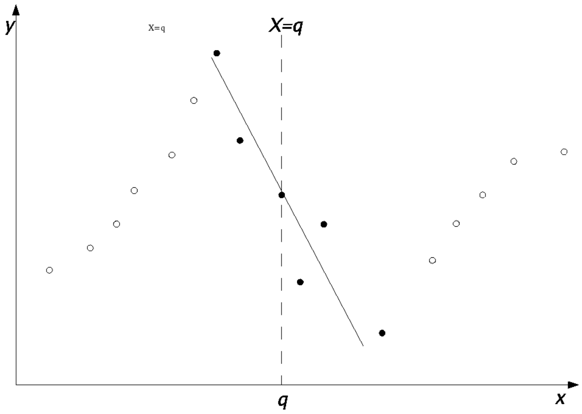

Fig. 3. Local linear model. The linear model estimates the value of the function in q by fitting the six nearest neighbors (bigger dots).

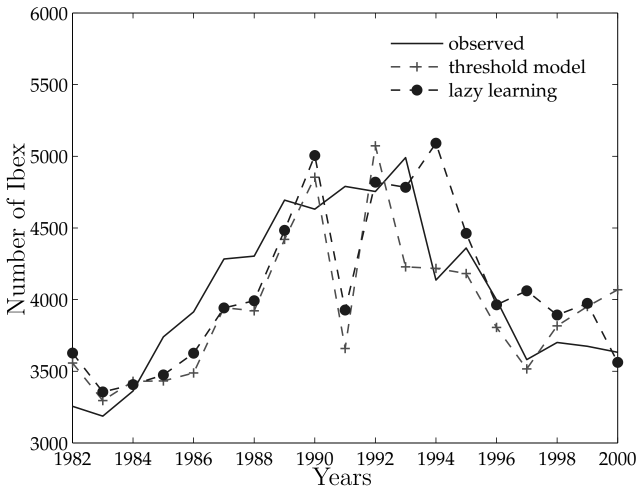

Fig. 4. Out-of-sample predictions computed by the threshold and the local model

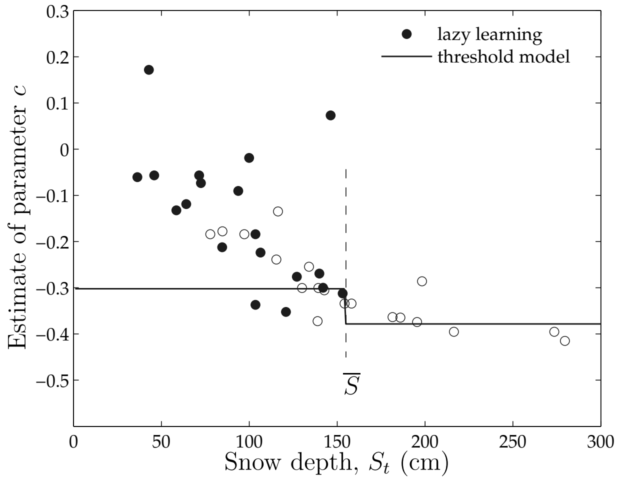

Fig. 5. Estimates of parameter c as a function of snow depth. Empty circles represent training data, while solid circles refer to testing values.