Mortality forecasting. Machine Learning Assisted Approach (MLAA)

Andrea Nigri

July 30, 2018

1 ABSTRACT

In the field of mortality, the Lee-Carter (L-C) based approach is the best way to forecast mortality rates. Since the first version of the model in 1992, scholars have developed different versions of it, so we could therefore define an "L-C model family" that includes all developments of it. Nevertheless, the first formulation of 1992, remains the benchmark for comparing the performance of future models. The main aim of this thesis is to try to fill a gap between demography methodology and statistical learning field. Indeed using data from Human Mortality Database we will attempt to integrate the L-C model with machine learning approach, in particular by using the Neural Network (NN).

2 INTRODUCTION

Mortality contributes significantly to population dynamics, indeed during the last two centuries developed countries have experienced a persistent increase in life expectancy[7]. Mortality modeling is crucial in economy, demography and social sciences, because through the mortality rates the prices of insurance products are determined and social policies are defined in the administrations. Considering the influence of mortality rates, it is necessary to model and forecast them in the near future. In order to predict future mortality rates, a model can be used to describe the mortality trend, for this purpose, scientists have always used statistical tools in this field. In recent years, thanks to the computational ability improvement, statistical learning techniques are back on stage. They were already known a few decades before and nowadays they are known as "machine learning". Unfortunately, scholars have always ignored the use of these techniques in the demographic field. Along this line of research, this paper attempts to bridge the gap between demography and machine learning, proposing new methods for investigating mortality processes. According to the demographic literature we have two types of mortality models[9]:

1. **Deterministic models:**

deMoivre(1725), Gompertz(1825), Makeham(1860), Weibull(1939).

2. **Stochastic models:** Alho(1990,1992), Alho and Spencer(1990), Bell and MOnsel (1990), Lee-Carter(1992).

The purpose of this paper is to get a Lee-Carter model estimation supported by Neural Network (NN). The first part is dedicated to the classic version of Lee-Carter model, in which we get the estimation of parameters and the final aim is not to give up the Lee-Carter model but integrate it with NN technique. In this sense, we can use the NN to forecast the "k" parameter in the model. The data comes from Human Mortality Database[4], after several trials, we chose the data from which we obtained the best forecasting performance, in particular the Danish male, from 1960 up to 2009.

3 MATERIAL and METHODS

3.1 Lee-Carter

The first approach to L-C model (1992) has been developed by the authors on U.S. mortality data, 1933-1987[5]. Trough the time several improvements have been made but the L-C 1992 remain still the benchmark for comparison with all future developments. The model can be written as[2]:

$$ln(m_{x,t}) = a_x + b_x k_t + \epsilon_{x,t} \tag{1}$$

with constraints:

$$\sum_x {b_x}^2 = 1; \sum_t k_t = 0 \tag{2}$$

Where $m_{x,t}$ are the observed central death rate at age x in year t, a_x is the average age-specific pattern of mortality, b_x is the pattern of deviations from the age of profile as the k_t varies. The parameter k_t is a time-trend index of general mortality level, forcasted using ARIMA with drift. The $\epsilon_{x,t}$ is the residual term at age x and time t. In their original paper, Lee and Carter (1992) applied a two-stage estimation procedure. In the first stage, singular value decomposition (SVD) is applied to the matrix of $log(m_{x,t}) - a$. Then in the second step, the time series of $k$is re-estimated by the method of so called "second stage estimation". According to the main aim of this thesis we will use a NN to forecast the k parameter in Lee Carter model. The model about k forecast can be written as:

$$k_t = f(k_{t-1}) + \epsilon_t \tag{3}$$

Where $k_{t-1} = (k_{t-1}, y_{t-2}, ..., k_{t-n})$ is a vector containing the values of the series and f is a neural network with n hidden nodes in a m layer. The error series ϵ is assumed to be homoscedastic.

3.2 Neural Network

The term neural network originated as a mathematical model inspired by the biological neural networks that constitute animal brains.[1] Indeed, every node of the graph represented a neuron, connected to each other by the arcs represented the synapses. A neural network is essentially a two-stage regression scheme, generally of nonlinear type. Following the graph in Figure 1 is possible formalize the neural network items. Each neuron in a network receives "weighted" information via these synaptic connections from the neurons that it is connected to and produces an output by passing the weighted sum of those input signals through an activation function $f(\alpha) = \frac{1}{1+e^{-\alpha}}$in the hidden layers. The net used in this analysis is the so-called "feed-forward neural networks" characterized by a lack of input-output interconnection between each neuron, in other words, there is no "feedback" from the outputs of the neurons towards the inputs. We indicate the generic input, latent,and output variables by x_j, H_n, and y_m, respectively.

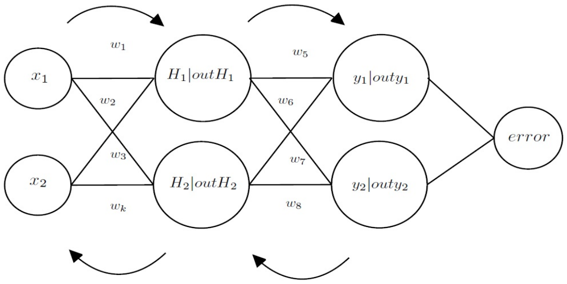


Figure 1: Schematical view of Neural Network: The circles represent neurons and lines represent synapses. Synapses take the input and multiply it by a "weight" (the "strength" of the input in determining the output). Neurons add the outputs from all synapses and apply an activation function.

[1] We now know that the animal brain is much more complex, but the term neural network survives[1].

3.2.1 BACKPROPAGATION

Training a neural network basically means calibrating all of the “weights” by repeating two key steps, forward propagation and back propagation:

- **In forward propagation** we apply a set of weights to the input data and calculate an output. For the first forward propagation, the set of weights is selected randomly.
- **In back propagation** we measure the margin of error of the output and adjust the weights accordingly to decrease the error.

Neural networks repeat both forward and back propagation until the weights are calibrated to accurately predict an output. Among many other learning algorithms, “back-propagation algorithm” is the most popular and the mostly used one for the training of feed-forward neural networks. It is, in essence, a means of updating networks synaptic weights by back propagating a gradient vector in which each element is defined as the derivative of an error measure with respect to a parameter The first step is the forward pass, necessary to carry out the backpropagation algorithm. The feed-forward pass consists of forwarding the input value in the hidden layer, summing the product of each input by its respective weight:

$$H_1 = x_1 w_1 + x_2 w_2 \tag{4}$$

More general:

$$H_n = \sum_{j=0}^{n} x_j w_j = w^T x \tag{5}$$

In each node, in every hidden layer (in the case of a multy layer NN), we obtain an output came from the sigmoidal activaction function:

$$outH_n = f(H) = \frac{1}{1 + e^{-H_n}} \tag{6}$$

The latter step is performed a number of times according to the number of hidden layers used. Then the output value from the hidden layer will be obtained, passing the hidden layer output to the input of the next layer through this following steps.

$$y_1 = outH_1 w_5 + outH_2 w_6 \tag{7}$$

$$y_m = \sum_{j=0}^{m} outH_j w_j \tag{8}$$

Finally is it possible to obtain the neural network output by applying activation function as usual.

$$out y_m = \frac{1}{1 + e^{-y_n}} \tag{9}$$

COST FUNCTION: Since the activation function is a continuous function it is differentiable. This property allows us to define a cost function[2] that can be minimized in order to update our weights.

$$e = (y - \widehat{y});\ J = \sum \frac{1}{2}(e)^2 \tag{10}$$

In order to minimize the cost function, we will use gradient descent[3].

For simplicity, let us consider a convex cost function for one single weight. We can describe the principle behind gradient descent as “climbing down a hill” until a local or global minimum is reached. At each step, we take a step into the opposite direction of the gradient, and the step size is determined by the value of the learning rate as well as the slope of the gradient.

w_k updated:

$$w_k.up = w_k - r\frac{\partial to\dot{t}alE}{\partial w_k} \tag{11}$$

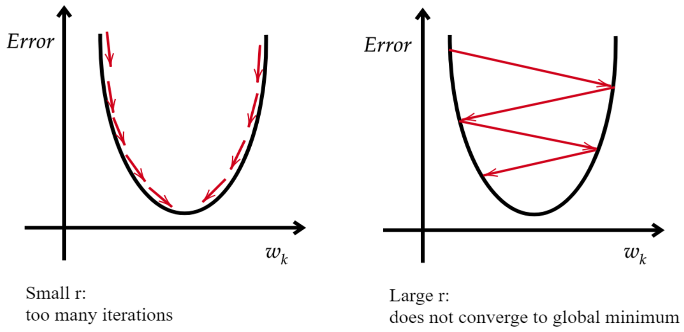
or

$$\Delta w = -r\nabla J(w) \tag{12}$$

[2] The fraction $\frac{1}{2}$ is just used for convenience to derive the gradient[8].

[3] A simple useful optimization algorithm used in machine learning to find the local minimum of linear systems[3].

The rate of change r is a hyperparameter set at 0.5 to ensure convergence, since a small value implies too many iterations while a large value does not allow convergence to the global minimum



Small r:
too many iterations

Large r:
does not converge to global minimum

Figure 2: Rate of change and gradient discending

DERIVATION STEPS: Derivation procedure using the chain rule applied in backpropagation

for $w_k; k = 5, ..., 8$

$$\frac{\partial totalE}{\partial w_k} = \frac{\partial totalE}{\partial out.y_k} \cdot \frac{\partial out.y_k}{\partial y_k} \cdot \frac{\partial y_k}{\partial w_k} \tag{13}$$

for $w_z; z = 1, ..., 4$

$$\frac{\partial totalE}{\partial w_z} = \frac{\partial totalE}{\partial out.H_n} \cdot \frac{\partial out.H_n}{\partial H_n} \cdot \frac{\partial H_n}{\partial w_z} \tag{14}$$

$$\frac{\partial totalE}{\partial out.H_n} = \frac{\partial E_1}{\partial out.H_n} + \frac{\partial E_2}{\partial out.H_{n+1}} \tag{15}$$

$$\frac{\partial E_1}{\partial out.H_n} = \frac{\partial E_1}{\partial y_k} \cdot \frac{\partial y_k}{\partial outH_n} \tag{16}$$

$$\frac{\partial E_1}{\partial y_k} = \frac{\partial E_1}{\partial outy_k} \cdot \frac{\partial outy_k}{\partial y_k} \tag{17}$$

4 RESULTS

We have applied the Lee-Carter model to mortality rates in time series from 1960 to 2009 in Danish male population. In order to explore the properties of the three components of the model, we plot them. As mentioned in the previous chapter, the parameter alpha represents the general age shape of mortality, in the same way, the b profile tell us which rates decline rapidly and which rates decline slowly in response to change in k.

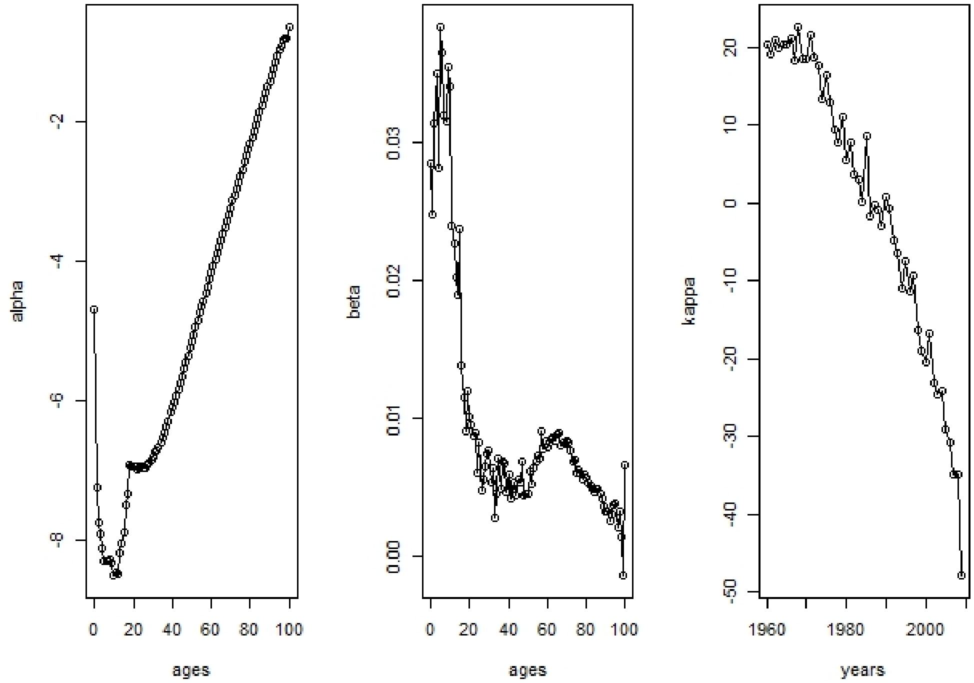


Figure 3: Lee Carter estimation

Reminding the main aim of this thesis we operate forecasting the k parameter. We carry out two procedure, in one hand we forecast the parameter using an ARIMA with drift, according to the classical scheme of Lee Carter. On the other hand, we will go on with the construction of the NN. For semplicity, we will omit the L-C details about forecasting and we will concentrate on NN technique. In order to build an NN is it necessary to specify the parameters: hidden layers and lag time. In this sense, data shall be divided into training data and testing data and we apply the NN model on it.

This training data has been used for learning while the testing data has

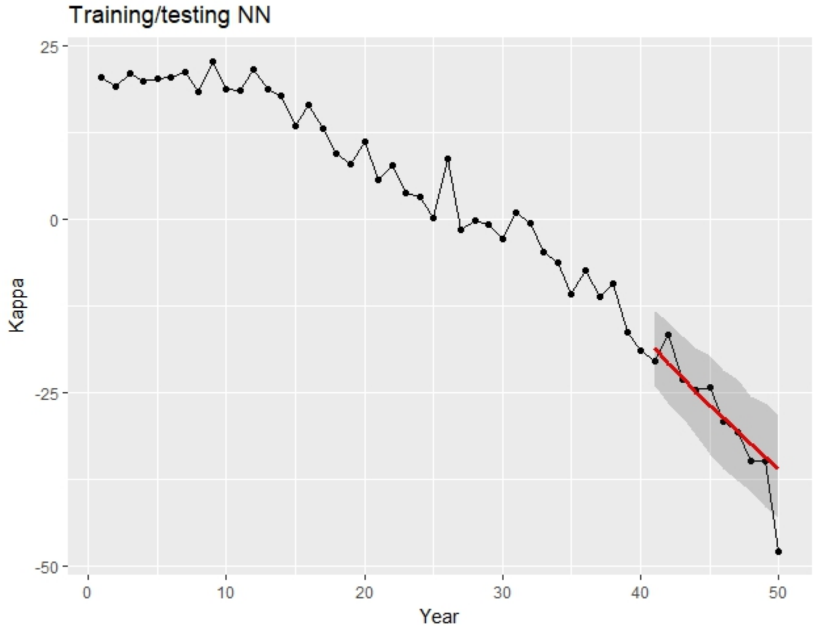


Figure 4: Training and testing Neural Network

been used to validate the model obtained, whether it is sufficient to describe the existing data or not. By using the last step we have formalized the NN's setting, in which we employ a single hidden layer with a lag of 5. Figure 4 shows a good forecast trend on the real data, validated by following values:

1. **Training set:** ME(-0.000714474), RMSE(2.783176)
2. **Test set:** ME(-1.024437346), RMSE(4.206276)

The next step is performing the trained model on the whole dataset and forecast the next 25 years. As we can see from the Figure 5, through the NN approach we can get a completely different result respect Lee Carter procedure. A Welch test has been performed to underline a significant statistical difference between the forecasted vector of k parameters (L-C Vs. NN: p.value < 0.01). In fact, a nonlinear estimation of the k parameter trend has been obtained, this characteristic has important reverberation on the forecast trend as well. Indeed as Figure 6 shows us, in the model comparison, they look meaningfully different from each other. The graphical differences have been confirmed from AIC values, in favour of NN method. Other important differences will be supported in the forecast of life expectancy at birth (Figure 7)

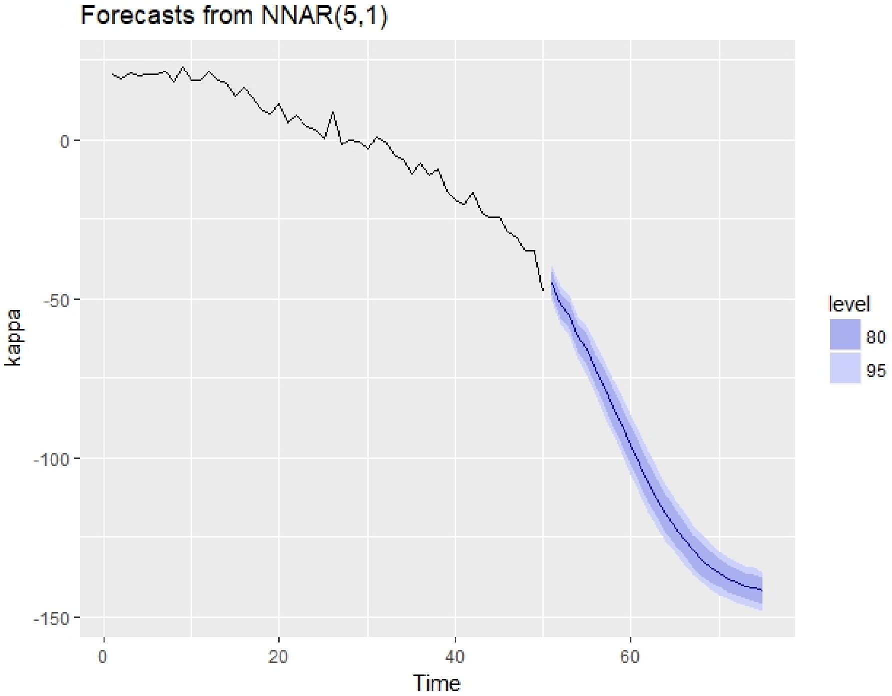


Figure 5: k-NN

5 CONCLUSION

The field of nonlinear estimation of k parameter in Lee Carter model, is in constant evolution. Nowadays is a common view among scholars thinking that a nonlinear behaviour of the k parameter could be a more plausible way to understand mortality patterns. Actually, the Lee-Carter forecast shape is likely too trivial to getting a realistic trend, in particular about life expectancy forecast. In this sense, the nonlinear shape make possible underline the differences in mortality trend, weighing in each of its parts (infant, adult and senescent components), the different shape in the mortality trend. In the analysis is evident how the employ of NN change the shape of future mortality rate (the last red part in the Figure 6), particularly into infant mortality section. Furthermore, we got the best evidence in life expectancy at birth's plot, in which the NN's power is more evident. In fact, through the nonlinear estimation, we can get an interesting life expectancy shape, very different from the classic L-C forecasting, which is in totally disagrees with the trend of last available data not forecasted by the model(from 2010 up to 2016, the black dots).

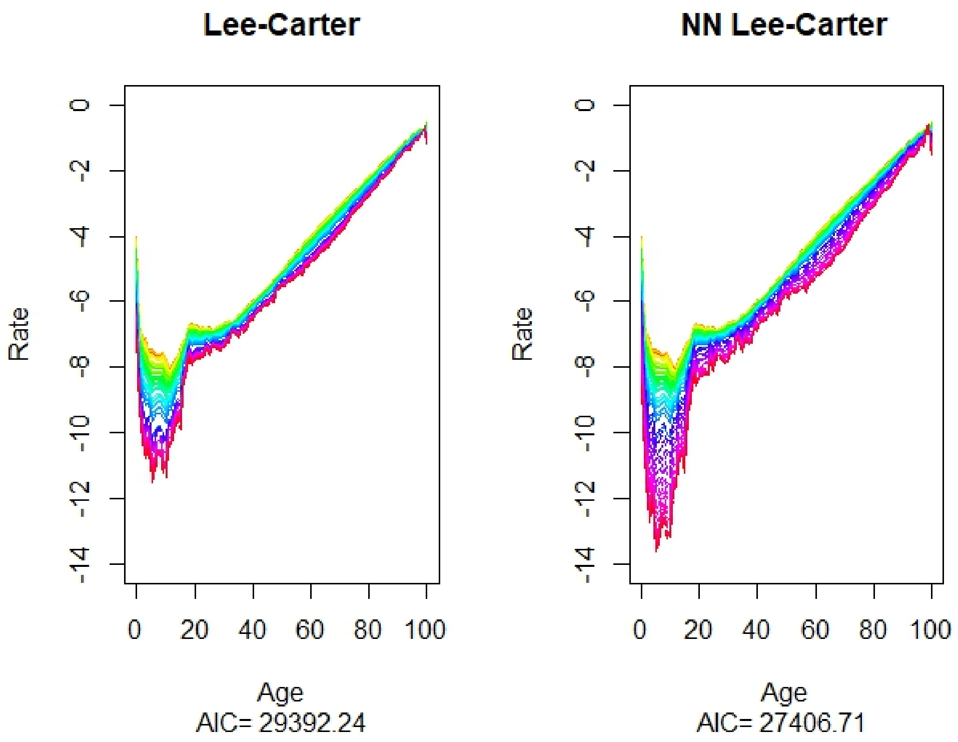


Figure 6: Lee Carter Vs. Neural Network

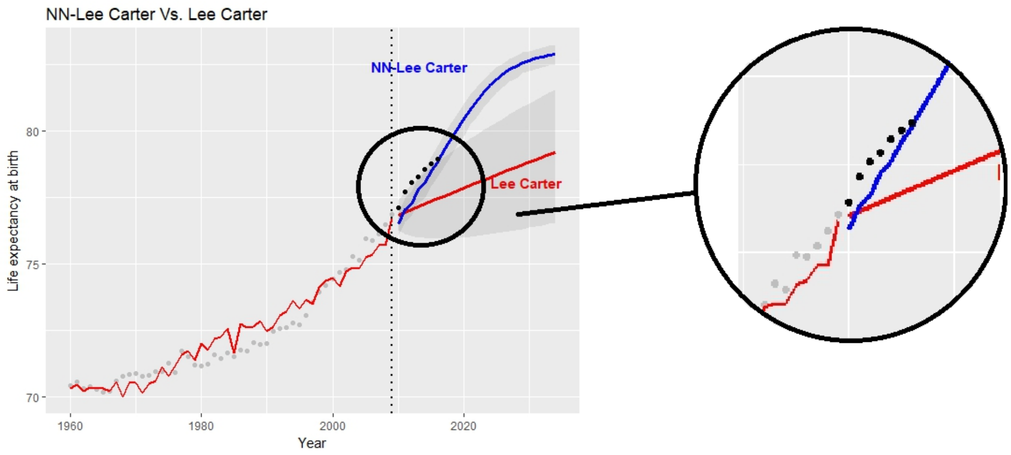


Figure 7: Life expectancy at birth: Lee Carter Vs. Neural Network

References

[1] Azzalini, A. Scarpa, B *Data Analysis and Data Mining: An Introduction.* 2012.

[2] Girosi, F. King, G. *Understanding the Lee-Carter Mortality Forecasting Method*. 2007.

[3] Hastie, T. Tibshirani, R and Friedman, J. *The Elements of Statistical Learning. Springer Series in Statistics.* 2009.

[4] Human Mortality Database *www.mortality.org.*

[5] Lee,R.D.,Carter, L.R. *Modeling and forecasting U.S.mortality. Journal of the American Statistical Association.* 1992.

[6] Murat, H. Sazli *A Brief Review Of Feed-Forward Neural Networks.* 2006.

[7] Oeppen, J., Vaupel, J. W. *Broken limits to life expectancy. Science.* 2002.

[8] Raschka, S *Python Machine Learning, 1st Edition.* 2015.

[9] Wang, J. Z. *Fitting and Forecasting Mortality for Sweden: Applying the Lee-Carter Model. Stockholm University.* 2007.