Correction to: High-Order Fuzzy Time Series Forecasting by Using Membership Values Along with Data and Support Vector Machine

Radha Mohan Pattanayak¹ · Sibarama Panigrahi² · H. S. Behera¹

Published online: 15 July 2020
© King Fahd University of Petroleum & Minerals 2020

Correction to: Arabian Journal for Science and Engineering
<https://doi.org/10.1007/s13369-020-04721-1>

In the original publication, ‘ALGORITHM 1’ was missed out completely during typesetting. It is given below:

In the original publication ‘ALGORITHM 1’ was unfortunately missed out completely. It is presented here. The original article has been corrected.

The original article can be found online at <https://doi.org/10.1007/s13369-020-04721-1>.

✉ H. S. Behera
hsbehera_it@vssut.ac.in

Radha Mohan Pattanayak
radhamohan.pattanayak@giet.ac.in

Sibarama Panigrahi
spanigrahi@suit.ac.in

¹ Department of Information Technology, Veer Surendra Sai University of Technology, Burla, Odisha 768018, India

² Department of Computer Science Engineering and Application, Sambalpur University Institute of Information Technology, Jyoti Vihar, Burla, Odisha 768019, India

---

Algorithm 1 Proposed fuzzy time series model with support vector machine (FTSF-SVM)

Input: Input a time series data as $y = [y_1, y_2, \dots, y_n]^T$

Output: Forecasting accuracy

Step1 : Determine the UOD for the time series data as $U = (D_{min} - D \quad D_{max} + D)$

Where the 'D' value is set to α times of sum of min_data and max_data value of the absolute first difference result of the sorted time series data. In the experiments, we have set $\alpha = 0.08$.

D_{min} and D_{max} are the lowest and highest values of time series data, respectively.

Suppose a time series $y = [8, 20, 4, 6, 1, 12, 4, 8, 15, 3, 6]$. After arranging the data in non-decreasing order and removing the duplicate data, the resultant time series will be $\hat{y} = [1, 3, 4, 6, 8, 12, 15, 20]$. The absolute first difference, $T_{diff} = [2, 1, 2, 2, 4, 3, 5]$. Then the value 'D' can be calculated as below.

$$\left\{ \begin{array}{l} T_{diff} = [abs(y_2 - y_1, y_3 - y_2, \dots, y_n - y_{n-1})] \\ \min\_data = \alpha * \min(T_{diff}) \\ \max\_data = \alpha * \max(T_{diff}) \\ D = \alpha * (\min\_data + \max\_data) \end{array} \right\}$$

Step2 : After finding the UOD, the number of intervals for the time series is computed as follows

$$NOI = \text{floor}(0.01 * \text{size}(\text{Timeseries}))$$

if ($NOI < 2$)

$$NOI = 2$$

end of if

Where NOI represents the number of intervals of the UOD.

Step3 : Determine the unequal-length of interval for the time series data by using fuzzy c-mean clustering method, which is implemented by using MATLAB function `fcm()` of MATLAB R2018b. By using `fcm()` function the cluster centre for each interval are calculated.

By considering the cluster centres, the lower and upper bound of each cluster is found by using the following equation

$$\left\{ \begin{array}{l} cl\_lb_m = U(1) \text{ if cluster is the first cluster i.e } m = 1 \\ cl\_ub_m = U(2) \text{ if cluster is the last cluster } m = l \\ cl\_ub_m = \frac{(cl_{c_m} + cl_{c_{m+1}})}{2} \\ cl\_lb_{m+1} = cl\_ub_m \end{array} \right\}$$

Where cl_lb , cl_ub and cl_c are considered as cluster lower bound, cluster upper bound and cluster centre, respectively for each cluster, $m = 1, 2, 3, \dots, l$ and m show the cluster number.

Step4 : Compute the membership values for each time series data by using triangular membership function.

for $i = 1$ to $\text{size}(\text{Timeseries})$

for $j = 1$ to NOI

if y_i belongs to $u_j(j, 1)$ and $u_j(j, 2)$

$$m_{ij} = \frac{(y_i - u_j(j, 1))}{(u_j(j, 2) - u_j(j, 1))}$$

else if y_i belongs to $u_j(j, 2)$ and $u_j(j, 3)$

$$m_{ij} = \frac{(y_i - u_j(j, 2))}{(u_j(j, 3) - u_j(j, 2))}$$

else

$$m_{ij} = 0$$

end of if

end of for

end of for

Step5 : Normalize the time series $y(t)$ to $y'(t)$ by using the min-max normalization technique.

$$y'_j(t) = \frac{(y_j(t) - \min_y)}{(\max_y - \min_y)}$$

Step6 : Determine the order of the fuzzy time series model (k) by using the ACF and PACF of time series.

Step7: Combine the normalized time series data $y'(t)$ with membership values and prepare the combined values.

Transform the data and membership values into $n-k$ patterns by using the sliding window technique. Each pattern in the pattern set consists of k -lagged data values along with their membership value as input, and the next value in the sequence is considered as the target value.

Step8 : Divide the pattern set into two categories, such as In-sample and Out-of-sample. Further, the In-sample data is partitioned into train and validation sets. The length of in-sample is set to 80% (train: 60% + validation: 20%) out-of-sample is set to 20% (test: 20%) of time series data.

Step9: Obtain the prediction result as $\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3 \dots \hat{y}_n]^T$ by optimizing the parameters of SVM. De-normalize to the prediction result i.e. $\hat{y}$ to get the actual forecasted value.

Step10: Compute the forecasting efficiency of the proposed model using two different error measures, namely root means square error (RMSE) and symmetric mean absolute percentage error (SMAPE) which are explained in Eq. 7 and 8.