IMPLEMENTING KOHONEN'S SOM WITH MISSING DATA IN OTB

Grégoire Mercier and Bassam Abdel Latif

Institut Telecom; Telecom Bretagne
CNRS UMR 3192 lab-STICC, team CID
Technopole Brest-Iroise,
CS 83818, F-29238 Brest Cedex 3, France

1. INTRODUCTION

This paper focuses on the implementation of the Kohonen's Self Organizing Map (SOM) through the Orfeo Toolbox. It makes the link between the theoretical background [1] and the strategic choices in the implementation:
1) Generic choice of the learning functions that ensure convergence of the training process;
2) Generic definition of the neuron that allows flexible use on multicomponent data, time series and also heterogeneous data;
3) Flexible class inheritance to redefine the update function to make the SOM deal with missing data.
Example will be given with the preprocessing of low resolution time series contaminated by clouds and shadows due to weather conditions during the acquisitions [2]. The technique is capable to reconstruct a complete time series free of clouds from MODIS data. It will also focus on the interest of a sparse distance criteria to reconstruct the time series on presence of haze that is difficult to detect.

2. KOHONEN'S SELF ORGANIZING MAP

The Kohonen's Self Organizing Map (SOM) is a neural network that defines a mapping from the input space onto a regular array of M nodes in one or two dimensions [3].

A reference vector, also called weight vector, $\boldsymbol{C}_m \in \mathbb{R}^n$ is associated to every node of position m on the map. An input vector $\boldsymbol{x} \in \mathbb{R}^n$ (to be processed or used during the training process) is to be compared to the $\boldsymbol{C}_m$. The best match is defined as output of the SOM: thus, the input data $\boldsymbol{x}$ is mapped onto this location (denoted to as $m_{\boldsymbol{x}}$).

The metric used to compare $\boldsymbol{x}$ to $\boldsymbol{C}_m$ is usually the Euclidean distance:

$$d\left(\boldsymbol{x}, \boldsymbol{C}_m\right) = \sum_{i=1}^{n} \left(x_i - C_{m;i}\right)^2 . \tag{1}$$

The node m that minimizes the distance between $\boldsymbol{x}$ and $\boldsymbol{C}_m$ defines the best-matching node (or the so-called winning neuron), and is denoted by the subscript $m_{\boldsymbol{x}}$:

$$d\left(\boldsymbol{x}, \boldsymbol{C}_{m_{\boldsymbol{x}}}\right) = \min_{m \in \{1,\ldots,M\}} d\left(\boldsymbol{x}, \boldsymbol{C}_m\right). \tag{2}$$

2.1. Training Phase

During the learning stage, the nodes which are close to the best-matching node will learn from the same input $\boldsymbol{x}$ also. While the initial values of the $\boldsymbol{C}_m$ may be set randomly, they will converge to a stable value at the end of the training process, by using:

$$\boldsymbol{C}_m(t+1) = \boldsymbol{C}_m(t) + h_{m,m_{\boldsymbol{x}}}(t)\Big[\boldsymbol{x}(t) - \boldsymbol{C}_m(t)\Big], \tag{3}$$

where t is the time parameter (*i.e.* the number of iterations). During one iteration of the training phase, every input $\boldsymbol{x}$, taken from a training set, is processed according to (3). $h_{m,m_{\boldsymbol{x}}}(t)$ is called *neighborhood kernel*: it is a function defined over the lattice points.

2.2. SOM Algorithm With Missing Values

SOM may be used to estimate, recover, missing values in a socio-economical database [4]. It can be used to estimate missing values in surveys [5]. In these two studies, observations are composed of a set of variables. Observations that have some missing variables were considered to as incomplete observations. In our application, MODIS data are collected in time series of reflectance (near-infrared and red reflectance's). In some other words, $\boldsymbol{x}$ is a temporal profile of a certain reflectance channel. If one temporal profile has cloud or shadow contamination in any date, then it is marked as having erroneous values. These erroneous values have to be detected and marked as missing values before proceeding with the SOM algorithm for missing values to recover them.

Observations, *i.e.* temporal profiles associated with each pixel, are supposed to be clustered into M classes of $\mathbb{R}^n$. When the input observation $\boldsymbol{x}$ is an incomplete vector, the set $\mathcal{M}_{\boldsymbol{x}}$ is introduced to define the indices of the missing values. $\mathcal{M}_{\boldsymbol{x}}$ is a sub-set of $\{1, 2, \ldots, n\}$. The winning neuron $\boldsymbol{C}_{m_{\boldsymbol{x}}}(t)$ related to $\boldsymbol{x}$ is usually founded by using (2) at iteration t. When facing incomplete vector $\boldsymbol{x}$, the distance $d\left(\boldsymbol{x}, \boldsymbol{C}_m(t)\right)$ is computed with the valid components of $\boldsymbol{x}$ only.

If incomplete observations are to be found in the training set, the update of the nodes (the best-matching one, $m_{\boldsymbol{x}}$, and its neighbors in $N_{m_{\boldsymbol{x}}}(t)$) affects the valid components only. By denoting $\boldsymbol{C}_m(t) = (C_{m;1}, \ldots, C_{m;k}, \ldots, C_{m;n})^t$ the components of vector $\boldsymbol{C}_m(t)$ and $\boldsymbol{x} = (x_1, \ldots, x_n)^t$, (3) becomes:

$$C_{m;k}(t+1) = \begin{cases} C_{m;k}(t) + h_{m,m_{\boldsymbol{x}}}(t)\Big[x_k - C_{m;k}(t)\Big] & \text{for } k \notin \mathcal{M}_{\boldsymbol{x}}, \\ C_{m;k}(t) & \text{otherwise.} \end{cases} \tag{4}$$

In fact, there is not need to detect the erroneous data before considering missing value when a sparse distance is used instead of eq. (1):

$$d\left(\boldsymbol{x}, \boldsymbol{C}_m\right) = \sum_{i=1}^{n} |x_i - C_{m;i}|^a \qquad \text{with } 0 \leqslant a \ll 1. \tag{5}$$

2.3. Estimation of Missing Values

Whatever the method used to deal with missing values, one of the most interesting properties of the algorithm is that it allows an *a posteriori* estimation of these missing values. Once the SOM has been trained, the missing values may simply be estimated by using:

$$\hat{x}_k = C_{m_{\boldsymbol{x}};k} \quad k \in \mathcal{M}_{\boldsymbol{x}}. \tag{6}$$

When the Kohonen's algorithm converges with a neighbor of length 0, it is known that the code-vectors $\boldsymbol{C}_m$ converges asymptotically to the mean value of its class m. Therefore, this estimation method consists in estimating the missing values of a random variable by the mean value of its class, defined through a training set. It is obvious that the more the compactness and the separability of the classes, the more accurate the estimation. Eq. (6) may be turned to a fuzzyfication by using membership values of the observation $\boldsymbol{x}$ to the set $\boldsymbol{C}$.

References

[1] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, September 1990.

[2] B Abdel Latif, R Lecerf, G Mercier, and L Hubert-Moy, "Low resolution time series analysis with erroneous data," *IEEE Trans. Geosci. Remote Sensing*, vol. 46, no. 7, July 2008.

[3] Kraaijveld, M.A. and Mao, J. and Jain, A.K., "A nonlinear projection method based on Kohonen's topology preserving maps," *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 548–559, May 1995.

[4] Marie Cottrell and Patrick Letrémy, "Missing values: processing whith Kohonen algorithm," in *International Symposium on Applied Stochastic Models and Data Analysis (ASMDA)*, Brest, France, 2005, pp. 17–20.

[5] Françoise Fessant and Sophie Midenet, "Self-Organising Map for Data Imputation and Correction in Surveys," *Neural Computing & Applications*, vol. 10, no. 4, pp. 300–310, 2002.