SOM and Feature Weights Based Method for Dimensionality Reduction in Large Gauss Linear Models

Fernando Pavón, fernando.pavon@gamco.es, GAMCO S.L. http://www.gamco.es

J. Vega, jesus.vega@ciemat.es Laboratorio Nacional de Fusion, CIEMAT, http://www.ciemat.es

Sebastián dormido Canto, sebas@dia.uned.es, Dpto. de Informática y Automática, UNED, http://www.dia.uned.es/

gamco

Content

gamco

1. Introduction

Discovering the most important variables is a crucial step for accelerating model building without losing potential predictive power of the data.

Dimensionality reduction can be a very critical issue in the data analysis of extremely large databases.

Nuclear fusion devices are experimental systems that produce large amounts of data

For example, JET is the largest nuclear fusion device in the world and collects O(10000) of signals per discharge (which means over 10 Gbytes of data per discharge). ITER will acquire about 500000 signals per discharge and this will correspond to over 1 Tbyte of data per discharge.

Objectives: to find out redundant information among the variables (linear/non-linear relationships), to weight the importance of each signal or variable and to find out some mechanism to reduce the amount of data.

$$Y_i = \theta_1 X_{i,1} + \theta_2 X_{i,2} + \ldots + \theta_n X_{n,1} + \varepsilon_i \qquad i = 1, \ldots, n$$

gamco

2. Variable Reduction

Variable reduction is a crucial step for accelerating model building without losing potential predictive power of the data.

The classical approximations to discover the importance of variables and the relationship among them:

- Graphical representation. It is simple: representation of each variable vs each other. If we have n variables, we need C(n,2) plots.
- Variable exploration using a tree (quasi brute-force). The method consists of exploring all the possible combinations of the variables.
- Linear correlation coefficients.

$$corrcoef(i,j) = \| \frac{\sum_{k=1}^{n}(x(i)_k - \bar{x}(i))(x(j)_k - \bar{x}(j))}{\sqrt{\sum_{k=1}^{n}(x(i)_k - \bar{x}(i))^2 \sum_{k=1}^{n}(x(j)_k - \bar{x}(j))^2}} \|$$

gamco

3. Feature Weighting Method Based on SOM

Our objective is to weight the different features, that is, each signal in the input Space.

A SOM is used to discover the most important features.

The proposed procedure to discover the most important signals is:

1. Generate a SOM with all the signals configured as inputs vectors. An input vector is made up of the sampled signals at a time t. The winner neuron is calculated using the Euclidean distance.

2. Considerate each neuron in the SOM like a class in a classification problem. We apply the feature weighted method using the trained SOM.

3. The signals which are linear or nonlinear combinations of others will have a large weight. The independent signals will have the lowest weights.

gamco

3. Feature Weighting Method Based on SOM

The Feature Weighting Method is:

Let $\xi_i^j = [\xi_{i1}^j, \xi_{i2}^j, ..., \xi_{id}^j]$, $i = 1, ..., N_j$ and $j \in (1, ..., N_n)$

The dimensional training sample of neuron *j*.

Where N_n is the number of neurons.

$N = \sum_{j=1}^{N_n} N_j$ is the total number of training samples.

In order to measure the difference among features, the global mean of total training samples of feature υ is calculated by

$$m_\upsilon = \frac{1}{N} \sum_{i=1}^{N_j} \sum_{j=1}^{N_n} \xi_{i\upsilon}^j, \qquad (\upsilon = 1, ..., d)$$

and local mean of class j of feature υ is

$$u_\upsilon^j = \frac{1}{N_j} \sum_{i=1}^{N_j} \xi_{i\upsilon}^j, \qquad (\upsilon = 1, ..., d)$$

gamco

3. Feature Weighting Method Based on SOM

The way to estimate the weights of features is adapted from LDA, the between-class variance S_v^B of the feature *v* which is defined by the following formulation

$$S_v^B = \frac{1}{N_c}\sum_{j=1}^{N_c}(m_v - u_v^j)^2, \qquad (v = 1, ..., d)$$

Similarly, the within-class variance S_v^J of the feature *v* is computed by

$$S_v^J = \sum_{j=1}^{N_n} P_j S_{jv}, \qquad (v = 1, ..., d)$$

Where $P_j = \frac{N_j}{N}$ is the prior probability of class j, and S_{jv} is the variance of class *j* of feature *v*. Finally, the weighting feature Ψ_v is the ratio of the between-class variance S_v^B to the within-class variance S_v^J, that is

$$\Psi_v = \frac{S_v^B}{S_v^J}, \qquad (v = 1, ..., d)$$

According to this rule, if the value of variance between classes is large, it implies that the distributions of classes are separate, and this feature is helpful for classification. On the other hand, if the value S_v^J is small, it indicates that inter class samples are close. Therefore, a larger weight should be given to this feature.

gamco

4. Experimental Results

$$Y_i = \theta_1 X_{i,1} + \theta_2 X_{i,2} + ... + \theta_n X_{n,1} + \varepsilon_i \qquad i = 1, ..., n$$

A synthetic data set has been used. 110 signals have been generated, and 10.000 values of each signal or variable.

The first 100 variables are random values from a uniform distribution, and the next 10 are linear and nonlinear combinations of the first 100.

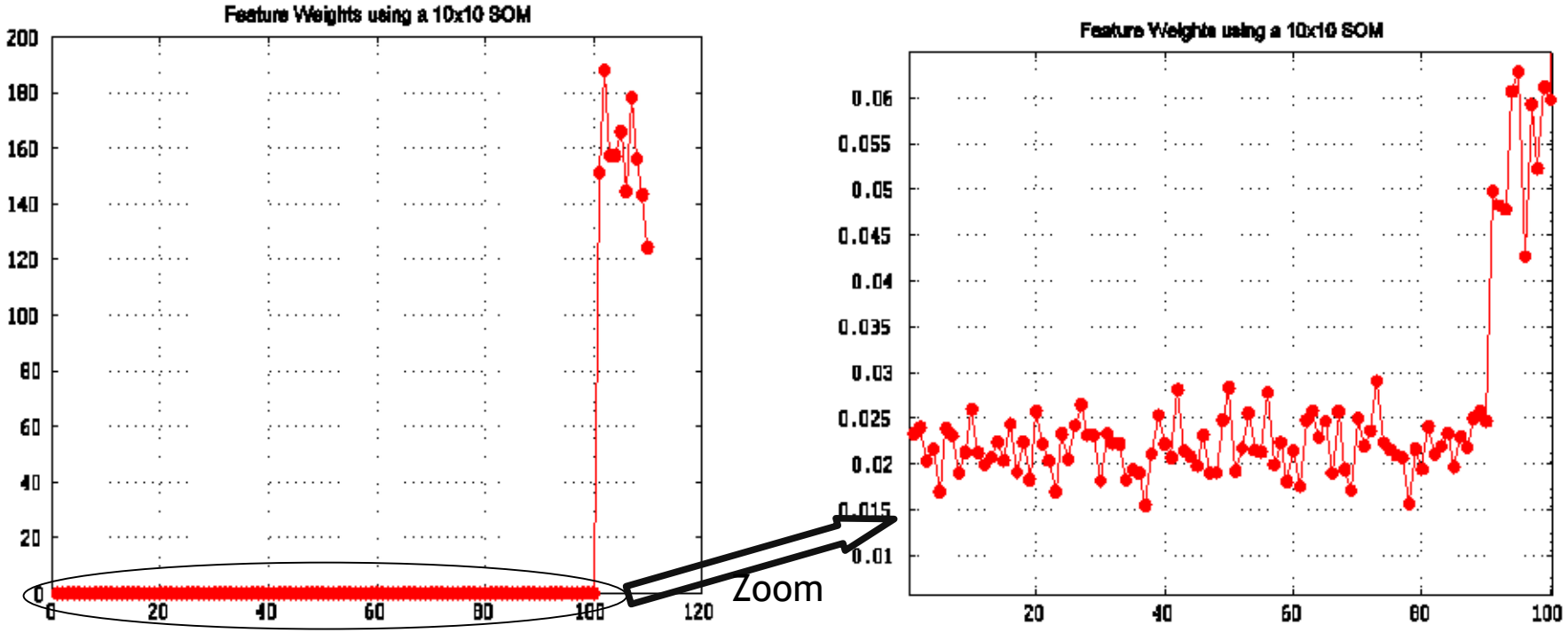
$$Signals = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,100} & Y_{1,1} & Y_{1,2} & \cdots & Y_{1,10} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,100} & Y_{2,1} & Y_{2,2} & \cdots & Y_{2,10} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,100} & Y_{n,1} & Y_{n,2} & \cdots & Y_{n,10} \end{pmatrix} \qquad \theta' s = \begin{pmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,10} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,10} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{100,1} & \theta_{100,2} & \cdots & \theta_{100,10} \end{pmatrix}$$

$$\theta_{1,j} ... \theta_{90,j} \in [0, 1] \; and \; \theta_{91,j} ... \theta_{100,j} \in [50, 100]; \; j = 1...10$$

Two sets of weights have been used: a *low values set* (between 0 and 1), and a *high values set* (between 50 and 100).

Using the synthetic data in the matrix signals we check the performance of the proposed method to reduce the number of variables by finding out the most relevant ones. Also, discovering redundant information.

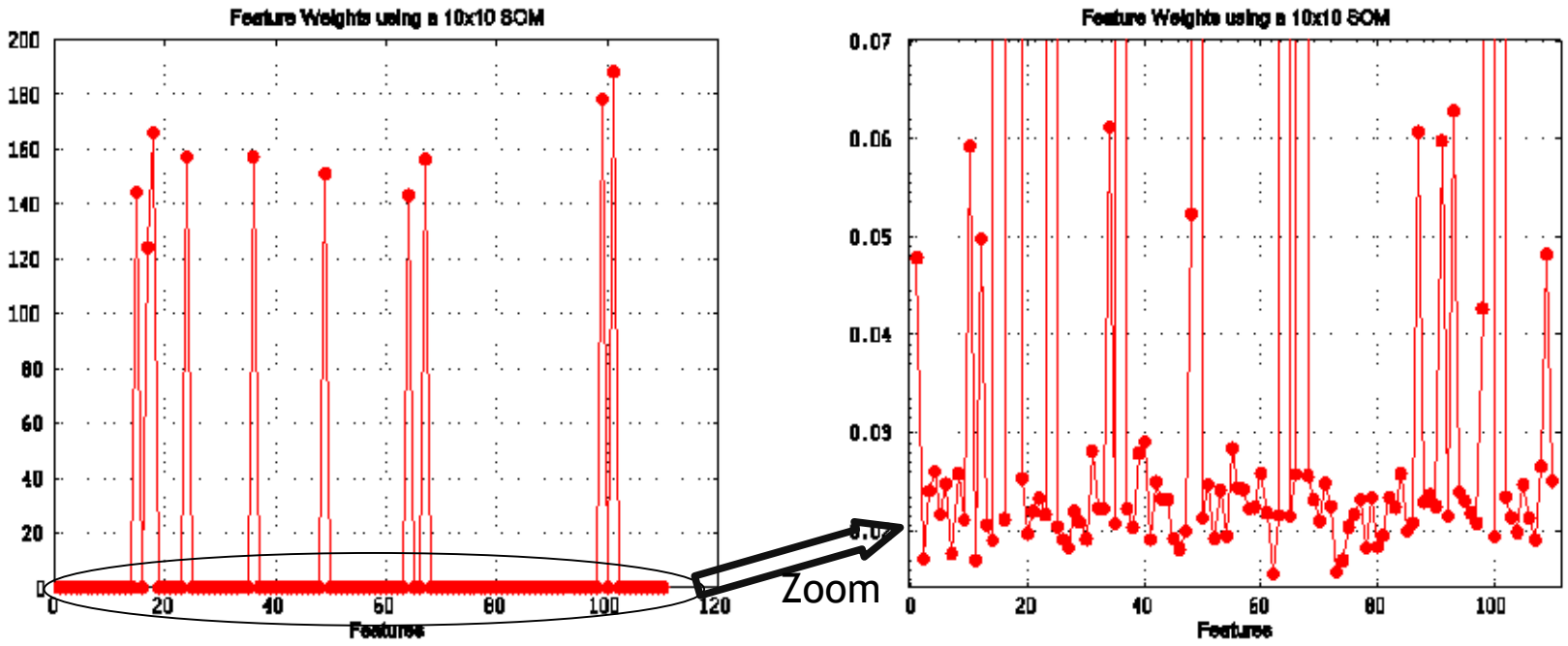
4. Experimental Results



The method of feature weighting identifies the variables which are combinations of others. Also the method discovers which variables *Xs* are most important ones

4. Experimental Results

The variables have been randomly permuted in their position in the input vector.



The capability of discovering the most important signals or the ones that are combination of others does not depend on the order of the variables in the input vectors.

gamco

5. Conclusion and Aplicabilities

In linear models, this method discovers dependant variables which are the result of linear and/or nonlinear combination of other variables. This method also discovers the importance of the variables for the linear regression. We can use the SOM networks in a first step to "split" the input vectors in the neurons and after this, to calculate the weight of each signal depending on their "within-class" and "between-class" variances.

This method can be applied to improve many areas, we will highlight two:

1. Classification and visualization problems using a weighted Euclidean distance for training a SOM.
2. Deep-learning techniques, in order to discover the most relevant characteristics for representation of a problem.
3. Linear models for data-driven approaches (physics based): L/H transitions, disruptions, plasma modelling, …

Tel.: (+34) 91 521 16 50
Fax: (+34) 91 522 99 84
c/ Alcalá 20, 4º·28014 Madrid
www.gamco.es