C. R. Mecanique 347 (2019) 754–761

Contents lists available at ScienceDirect

Comptes Rendus Mecanique

www.sciencedirect.com

Data-Based Engineering Science and Technology / *Sciences et technologies de l'ingénierie basées sur les données*

Code2vect: An efficient heterogenous data classifier and nonlinear regression technique

Clara Argerich Martín [a], Ruben Ibáñez Pinillo [a], Anais Barasinski [b], Francisco Chinesta [c,*]

[a] *PIMM, Arts et Métiers Institute of Technology, CNRS, CNAM, HESAM University, 151, boulevard de l'Hôpital, 75013 Paris, France*
[b] *University of Pau & Pays Adour, E2S UPPA, IPREM UMR5254, 64000 Pau, France*
[c] *ESI GROUP Chair @ PIMM, Arts et Métiers Institute of Technology, 151, boulevard de l'Hôpital, 75013 Paris, France*

ARTICLE INFO

Article history:
Received 20 May 2019
Accepted 1 August 2019
Available online 15 November 2019

Keywords:
Machine learning
Data representation
Classification
Categorial data
Neural network
High-dimensional data
Regression

ABSTRACT

The aim of this paper is to present a new classification and regression algorithm based on Artificial Intelligence. The main feature of this algorithm, which will be called Code2Vect, is the nature of the data to treat: qualitative or quantitative and continuous or discrete. Contrary to other artificial intelligence techniques based on the "Big-Data," this new approach will enable working with a reduced amount of data, within the so-called "Smart Data" paradigm. Moreover, the main purpose of this algorithm is to enable the representation of high-dimensional data and more specifically grouping and visualizing this data according to a given target. For that purpose, the data will be projected into a vectorial space equipped with an appropriate metric, able to group data according to their affinity (with respect to a given output of interest). Furthermore, another application of this algorithm lies on its prediction capability. As it occurs with most common data-mining techniques such as regression trees, by giving an input the output will be inferred, in this case considering the nature of the data formerly described. In order to illustrate its potentialities, two different applications will be addressed, one concerning the representation of high-dimensional and categorical data and another featuring the prediction capabilities of the algorithm.

1. Introduction

The use of artificial intelligence to work with data in order to classify, represent or analyze it is a field that is attracting the more and more interest from both the scientific and technological viewpoints. Many techniques for this purpose have been developed during the last years, such as the Locally Linear Embedding, presented in [1] and largely used for diverse applications [2]. The tSNE (t-distributed Stochastic Neighbor Embedding) presented in [3], is widely used for visualizing high-dimensional data and discover the latent manifold in which data is embedded, whose dimensionality informs on the number of explicative uncorrelated parameters. Other algorithms were designed for extracting hidden regressions from data, and then being able to infer a response (output) for a given input. Linear and nonlinear regressions were widely considered in a variety of works and diverse applications, decision trees and random forest were proposed for operating in highly

[*] Corresponding author.
E-mail addresses: clara.argerich_martin@ensam.eu (C. Argerich Martín), Ruben.IBANEZ-PINILLO@ensam.eu (R. Ibáñez Pinillo), anais.barasinski@univ-pau.fr (A. Barasinski), Francisco.CHINESTA@ensam.eu (F. Chinesta).

https://doi.org/10.1016/j.crme.2019.11.002

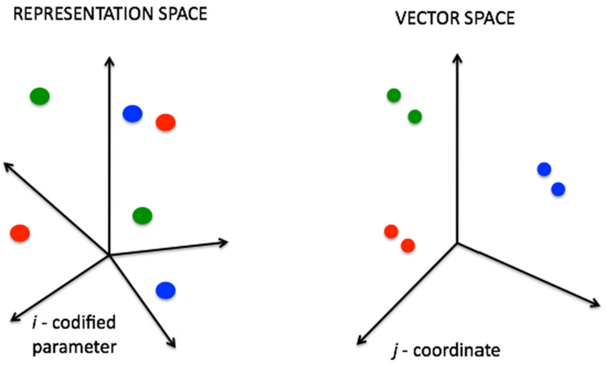


Fig. 1. Input space (left) and target vector space (right).

dimensional spaces. In [4] authors perform a review of these techniques and some applications addressed in [5] using random forests. Regression techniques operate quite well even in the low-data limit, in particular the ones based on sparse sensing combined with the use of separated representation formats. When data is abundant, the most powerful learning techniques are the ones based on the use of neural networks, given place to the so-called deep-learning (see [6] and [7] for general knowledge on deep learning).

One of the most used applications of deep Learning is visual recognition and image classification [8,9]. Deep learning techniques are also used for probabilistic language [10]. Interested readers can refer to [11] and the numerous references therein.

One of the most recent algorithms developed for heterogeneous data classification is the so-called Word2Vec, developed by Google inc. This algorithm presented in [12] is based on the use of neural networks [13] and the representation of each word as a vector representing the probability of having a given neighboring word. This approach tried to circumvent the difficulty of defining distances between qualitative data, that is, one could consider that yellow and red are quite close because both are representing colors, however, from the spelling of both words such a proximity becomes difficult to quantify.

In fields like manufacturing engineering such an approach is particularly relevant because the performances of for example a manufactured plastic part depends on the resin used (defined by its commercial appellation or the chemical nature also codified but differently), the oven temperature (in a particular unit system), the curing time and the name of the employe (e.g., Peter, Paul...).

Now, a process could be characterized from these four parameters: (i) resin, (ii) temperature, (iii) process time and (iv) employe surname; and with each one we could associate a label associated with a particular output of interest, a value quantifying a particular performance of the formed part.

Thus, each process could be described by a point in a four-dimensional space, each point having a color depending on the value associated with the output of interest. The four axes represent the heterogeneous parameters appropriately codified. For example, a numeric code must be associated with each resin or operator name (e.g., polyester = 1, vinylester = 2,... Peter = 1, Paul = 2,...) in order to represent qualitative data on the resin or operator axes...

However, these choices are totally arbitrary, as well as units in the different quantitative data (temperature and process time in our example). Thus, it is evident that calculating distances between processes in such a space has not sense. In other words, two points representing two processes could be very close in the four-dimensional space but have outputs of interest very different and, oppositely, remote points could imply very similar performances.

This issue is not new, and many authors when considering heterogeneous but quantitative data, normalized them for minimizing the impact of units in the computation of distances (for example usual stresses and strain in solid mechanics differ in about 12 orders of magnitude, the former being of order 10^6 and the last 10^{-6}). However, normalization hides the issue without really solving it.

In what follows, we propose a technique, sketched in Fig. 1, for mapping points from a representation space to a target space equipped of an Euclidean metric allowing the quantification of distances and then applying all the numerical artillery using distances, as considered later.

2. Classification and regression strategies

In this section, the methodology sketched in Section 1 is described. We assume that the points in the original space (space of representation) consist of $\mathtt{P}$ arrays composed on N entries (four in the example considered above). They are assumed arrays because they cannot be considered vectors, and are noted by $\mathbf{y}_i$. Their images in the vector space are noted by $\mathbf{x}_i \in \mathbb{R}^d$, with $d \ll N$, this time real vectors subjected to the rules of coordinate transformation. That vector space is equipped with the standard scalar product and the associated Euclidean distance. The mapping is described by the $d \times N$ matrix $\mathbf{W}$,

Table 1
Codification of categorial parameters.

Param. 1	Code	Material	Code	Supplier	Code	Param. 2	Code	Stable	Code
A	10	*A*	1	*A*	1.5	*A*	7	*YES*	1
B	11	*B*	2	*B*	2.5	*B*	8	*NO*	0.1
C	12	*C*	3	*C*	3.5				
D	13	*D*	4	*D*	4.5				
E	14	*E*	5						
F	15	*F*	6						
G	16								

$$\mathbf{x} = \mathbf{W}\mathbf{y}, \tag{1}$$

where both the components of $\mathbf{W}$ and the images $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \cdots, \mathtt{P}$ must be calculated. Each point $\mathbf{x}_i$ keeps the label, denoted by $\mathcal{O}_i$ (value of the output of interest, here assumed scalar), associated with its origin point $\mathbf{y}_i$.

We would like placing points $\mathbf{x}_i$, such that the Euclidean distance with each other point $\mathbf{x}_j$ scales with their outputs difference, i.e.

$$(\mathbf{W}(\mathbf{y}_i - \mathbf{y}_j)) \cdot (\mathbf{W}(\mathbf{y}_i - \mathbf{y}_j)) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = |\mathcal{O}_i - \mathcal{O}_j|, \tag{2}$$

where the coordinates of one of the points can be arbitrarily chosen. Thus, there are $\frac{\mathtt{P}^2}{2} - \mathtt{P}$ relations for determining the $d \times N$ (the components of $\mathbf{W}$ because the coordinates $\mathbf{x}_i$ are obtained from $\mathbf{W}$ by applying relation (1)).

Linear mappings are limited and do not allow operating in nonlinear settings. Thus, a better choice consists of the nonlinear mapping $\mathbf{W}(\mathbf{y})$, expressible from the general polynomial form

$$\mathbf{W}(\mathbf{y}) = \sum_{k=1}^{\mathtt{K}} \mathbb{W}_k \mathcal{P}_k(\mathbf{y}), \tag{3}$$

where matrices $\mathbb{W}_k$ have the same dimensions as $\mathbf{W}$ and must ensure the $\mathbf{y}_j \rightarrow \mathbf{x}_j$ mapping.

On the other hand, the interpolation functions $\mathcal{P}_k(\mathbf{y})$ can be continuous or discontinuous. In the numerical applications described later, standard polynomial or radial functions are considered.

The associated nonlinear problem can be efficiently solved by using an adequate linearization strategy, e.g., Newton's method as presented in Appendix A. Because of the increase in the number of unknowns to be determined, a rich enough sampling is required.

The choice of d deserves some comments. The lower d is, the stronger becomes the mapping nonlinearity, however the visualization becomes simpler.

When the dimensionality N increases approximations suffer the so-called curse of dimensionality. A very efficient way of alleviating it consists in using separated representations [14] from which the mapping reads

$$\mathbf{W}(\mathbf{y}) = \sum_{k=1}^{\mathtt{K}} \mathbb{W}_k \prod_{i=1}^{N} P_k^i(y_i). \tag{4}$$

In the previous expression y_i refers to the i-component of $\mathbf{y}$.

The use of these separated representations constitutes a work in progress.

3. Numerical examples

This section addresses two case studies, the first devoted to classification of heterogenous data and the second aiming at emphasizing the classification and nonlinear regression capabilities of the technique just introduced.

3.1. Representation of high-dimensional categorical data from manufacturing processes

In this section an application with real and highly non-linear data is presented. We consider a database provided by an industry related to a real manufacturing process. The input space concerns six manufacturing parameters collected during 53 realizations of the process. Parameters concern four categorical parameters: material, supplier, two process parameters whose quantification is shown in Table 1 and two other quantitative process parameters. The output is the stability of the process itself, also quantitively codified as shown in Table 1. Words are codified by using numbers into the $[-1, 1]$ interval.

In this particular case the input data, $\mathbf{y}_i$, $\forall i$, is grouped in a 6×53 matrix (53 realizations of the process defined by 6 input parameters), with a single scalar target. The vector space $\mathbf{x}_i$ consists of a 2×53 matrix (53 realizations mapped into the low-dimensional space $d = 2$).

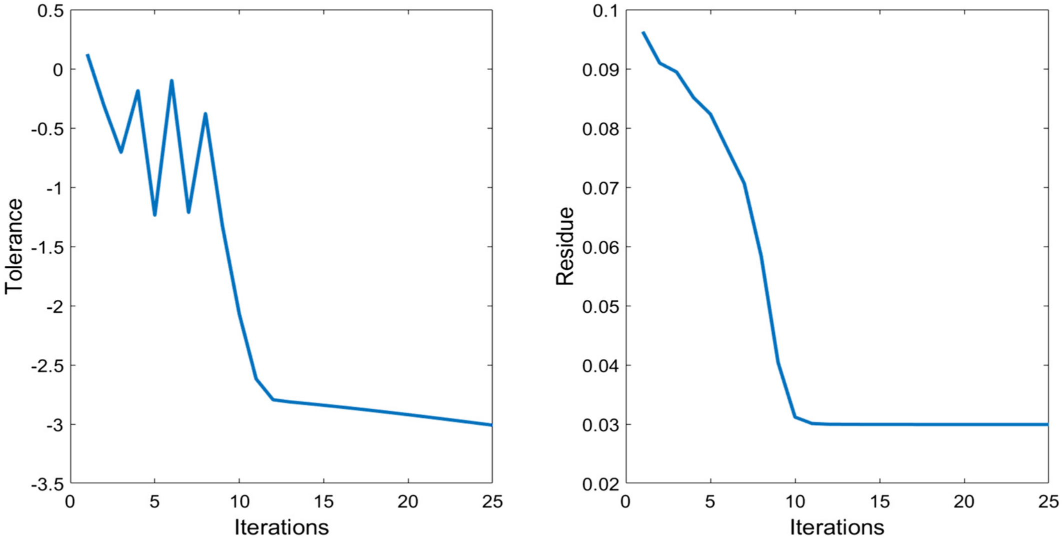
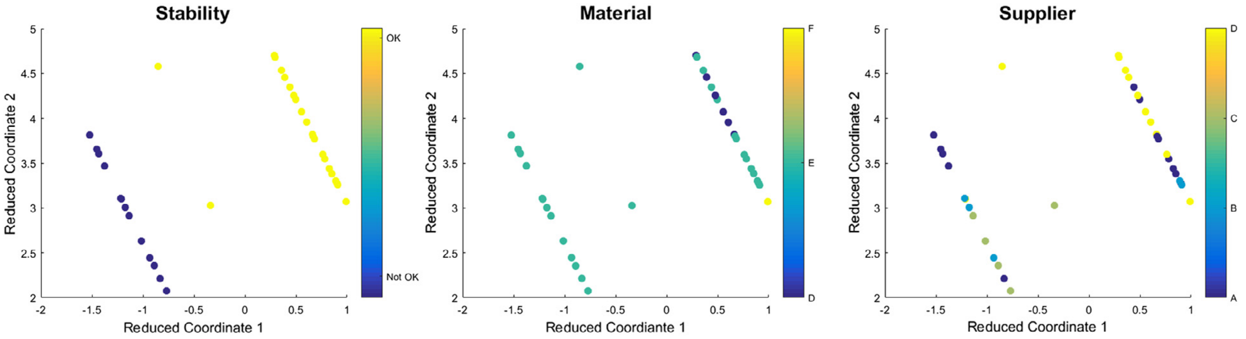


Fig. 2. Residue and tolerance related to the computation of mapping **W**.



(a) Clusters colored by Output (b) Clusters colored by Material (c) Clusters colored by Supplier

Fig. 3. Vector space representations: output, material, and supplier.

Once the input data and the output is quantitatively codified, we run the algorithm described in the previous section, in particular its nonlinear counterpart, that reaches convergence in few iterations (indicated by index i) as Fig. 2 proves, where the tolerance, defined by

$$\text{Tolerance} = \log \frac{\|\mathbf{W}_i - \mathbf{W}_{i-1}\|}{\|\mathbf{W}_{i-1}\|} \tag{5}$$

is depicted at each iteration, with the residue of the Newton iteration (8).

Fig. 3(a) depicts the mapped samples in the vector space, that is $\mathbf{x}_i$, where the 53 points representing each realization of the manufacturing process have been colored according to its output. As the reader can appreciate, two well differentiated clusters are obtained, one concerning all the stable processes and the other grouping non-stable conditions, with only two points out of the clusters. It is important to note that better results can be obtained by increasing both the sampling and the approximation of the nonlinear mapping, that is $\mathtt{K}$ in the approximation (3).

Other than clustering with respect to the output, or using the computing mapping for inferring stability for new process conditions, one could also try to extract the relevance of the different parameters on the output. For that it suffices coloring the mapped points according to the value of the considered parameter. Figs. 3(b) and 3(c) report this analysis for both material and supplied respectively. From these images it can be concluded that materials F and D and supplier D exhibit stability. On the contrary, supplier C compromises process stability.

The same analysis was performed by using NN (neural network) with a hidden layer composed of two neurons whose training was ensured by data $\mathbf{y}_i$ and the associated stability outputs. The NN is shown in Fig. 4, with 53 data consisting of 6 inputs, with a single output. The dimension is reduced from 6 to 2 by using 2 neurons in the hidden layer, for emulating the low-dimensional space generated by *Code2Vect*, which is depicted in Fig. 5 where points associated with stable process conditions are colored in blue. It can be concluded that clustering is not achieved. Even if we cannot conclude on the

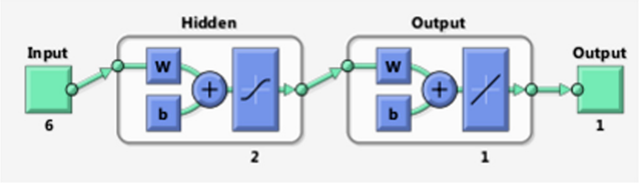


Fig. 4. Neural network.

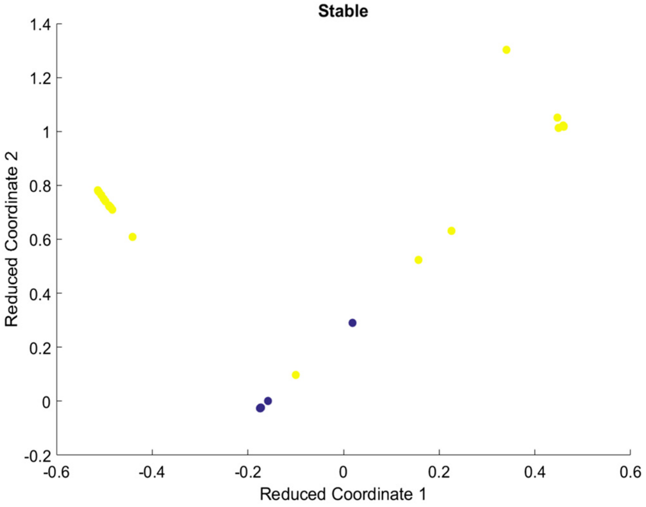
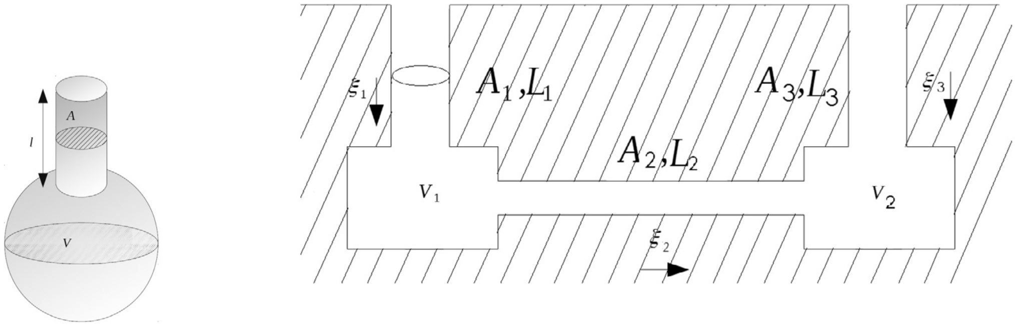


Fig. 5. NN-based model.



(a) Helmholtz resonator

(b) Acoustic system: the solid is represented by parallel lines and the air flows through the three channels connecting the two cavities.

Fig. 6. Acoustic absorbers.

superiority of one procedure with respect to the other (because the large number of NN variants), we can conclude on the simplicity and efficiency of the *Code2Vect* proposal.

3.2. *Model and regression construction*

We consider acoustic systems composed by acoustic resonators, whose simplest structure, the Helmholtz resonator – HR – is depicted in Fig. 6a, and involves three main geometric parameters, length (L), section (A) and volume (V). HR can be combined to create more complex acoustic structures as illustrated in Fig. 6b, whose behavior is governed by the dynamic system $\mathbf{M}\ddot{\xi}+\mathbf{C}\dot{\xi}+\mathbf{K}\xi=f$ (where ξ defines the air motion amplitude at every bottleneck) and was deeply analyzed in [15], from which the resonance frequencies can be extracted.

Without loss of generality, the acoustic system consider: $L_1=L_2=L_3$, $A_1=A_2=A_3$ and $V_1=V_2$, reducing the number of input parameters to three, like in the case of the simple Helmholtz resonator. Now 64 different geometrical combinations

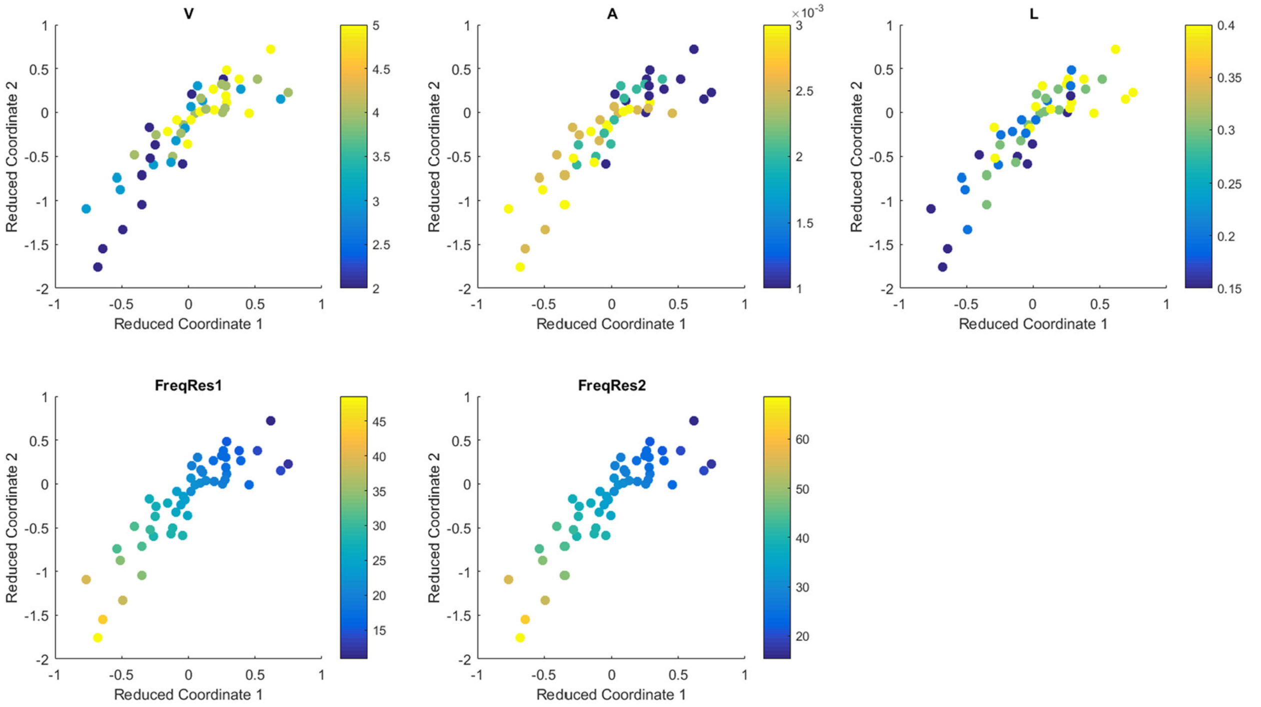


Fig. 7. Vector space colored with respect to the parameter values (top) and output of interest (bottom).

were considered and the two resonance frequencies computed, both constituting the output to be predicted by the searched data-based model.

Results obtained when applying the *Code2Vect* technique are depicted in Fig. 7, where the two images at the bottom represent the vector space with samples colored according to the associated outputs. In the present case, the scalar output considered in the mapping construction (Eq. (8)) consists of the norm of the vectors containing both frequencies, i.e. with $\mathbf{t}_i^\mathsf{T} = (\omega_i^1, \omega_i^2)$, $|\mathcal{O}_i - \mathcal{O}_j| = \|\mathbf{t}_i - \mathbf{t}_j\|$.

From Fig. 7, we can conclude that section and volume (V and L) are inversely correlated with the output, because for example yellow color concentrates on the bottom when representing the output whereas it localizes on the top when referring to these parameters. The opposite occurs when considering the cross-section area (A). These findings are in perfect agreement with the known physics of a Helmholtz Resonator, whose resonance frequency ω_r is given by

$$\omega_\mathrm{r} = \frac{c}{2\pi}\sqrt{\frac{A}{VL}} \tag{6}$$

Now, with the model $\mathbf{W}(\mathbf{y})$ relating geometrical parameters and resonance frequencies extracted, its predictive capabilities are being tested. For that purpose the model is extracted by using only 60% of the 64 geometrical variants and the rest of the samples (the remaining 40% $\mathbf{y}_j$) used for validating the predictions, that is, for comparing the predictions, the output $\mathbf{t}_i$ at $\mathbf{x}_j = \mathbf{W}(\mathbf{y}_j)\mathbf{y}_j$ interpolated from the output $\mathbf{t}_i$ at the neighboring samples $\mathbf{x}_i$.

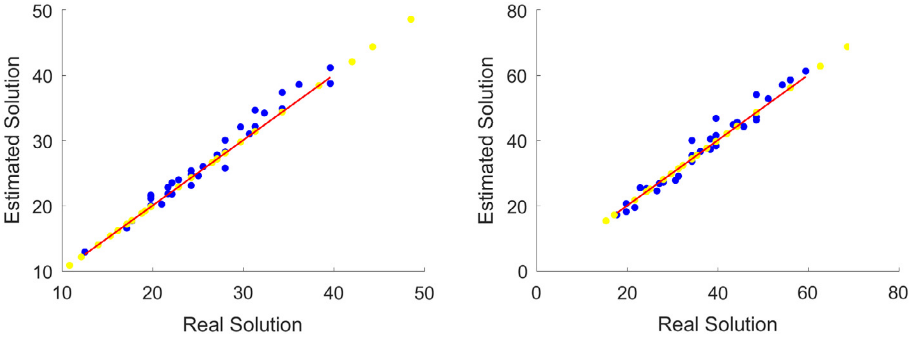
The predictions are depicted in Fig. 8; in yellow the samples used as training and in blue the predicted. The deviation with respect to the diagonal represents the prediction error that results, as expected, very low for the samples used in the training and a bit larger, but lower than 4% for the others.

From the discussion above, one expects that the best model is the one relating the combined parameter A/VL with the resonance frequencies, instead of the one that we considered above that considered the geometrical parameters individually, that is, L, A, and V. In order to prove the superiority of modeling by considering the best parametrization, Fig. 9 compares the previous prediction with the one obtained by using the combined parameter A/VL that produced almost perfect predictions.

However, the physics behind data is not always known to inform the best choice of parameters. A procedure for accessing to those most relevant combined parameters constitutes one of our main research activities still in progress. For the moment, and in absence of “a priori” information, parameters will be selected and incorporated individually in the model.

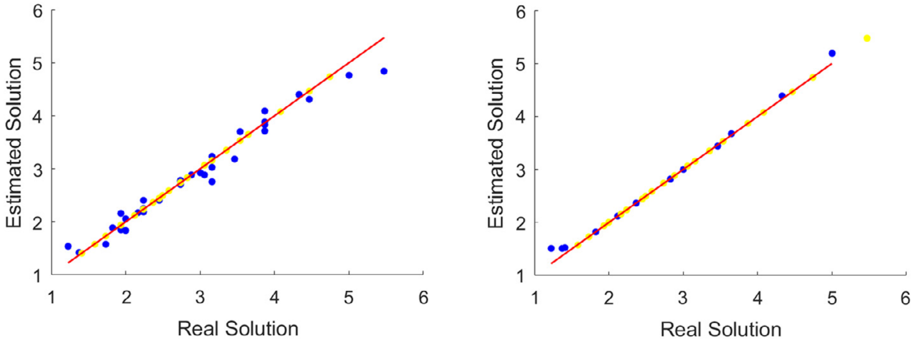
4. Conclusions

The classification algorithm presented here enables the representation of high-dimensional heterogeneous data, including continuous and discrete, quantitative and categorial. It establishes a mapping between a representation space and a vector



Prediction of the first ressonance frequency Prediction of the second ressonance frequency

Fig. 8. Prediction of the target.



Prediction using the three geometrical parameters Prediction using the combined parameter

Fig. 9. Predicting the output.

space that is kept low dimensional for facilitating visualization. It allows supervised classification but also making prediction from its nonlinear regression nature. Moreover, it enables a very simple representation emphasizing the relevance of the parameters involved in the model. Its performances in the low-data limit have been proved.

Acknowledgements

The authors gratefully acknowledge the ANR ("Agence nationale de la recherche", France) for its support through project AAPG2018 *DataBEST*. AIRBUS Group through the PhD CIFRE program and ESI Group through the ESI-ENSAM Chair program.

Appendix A. Nonlinear modeling

By denoting $\mathbf{W}_i = \mathbf{W}(\mathbf{y}_i)$, the Newton proceeds at iteration n

$$\mathbf{W}_{n+1,i} = \mathbf{W}_{n,i} + \Delta\mathbf{W}_i, \tag{7}$$

that replaced in the nonlinear counterpart of Eq. (2)

$$(\mathbf{W}_i\mathbf{y}_i - \mathbf{W}_j\mathbf{y}_j)\cdot(\mathbf{W}_i\mathbf{y}_i - \mathbf{W}_j\mathbf{y}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 = |\mathcal{O}_i - \mathcal{O}_j|, \tag{8}$$

the left hand member reads

$$\begin{aligned}&(\mathbf{y}_i^{\mathsf{T}}(\mathbf{W}_{n,i}^{\mathsf{T}} + \Delta\mathbf{W}_i^{\mathsf{T}}) - \mathbf{y}_j^{\mathsf{T}}(\mathbf{W}_{n,j}^{\mathsf{T}} + \Delta\mathbf{W}_j^{\mathsf{T}}))((\mathbf{W}_{n,i} + \Delta\mathbf{W}_i)\mathbf{y}_i - (\mathbf{W}_{n,j} + \Delta\mathbf{W}_j)\mathbf{y}_j) = \\ &\mathbf{y}_i^{\mathsf{T}}(\mathbf{W}_{n,i}^{\mathsf{T}} + \Delta\mathbf{W}_i^{\mathsf{T}})(\mathbf{W}_{n,i} + \Delta\mathbf{W}_i)\mathbf{y}_i + \mathbf{y}_j^{\mathsf{T}}(\mathbf{W}_{n,j}^{\mathsf{T}} + \Delta\mathbf{W}_j^{\mathsf{T}})(\mathbf{W}_{n,j} + \Delta\mathbf{W}_j)\mathbf{y}_j - \\ &2\mathbf{y}_j^{\mathsf{T}}(\mathbf{W}_{n,j}^{\mathsf{T}} + \Delta\mathbf{W}_j^{\mathsf{T}})(\mathbf{W}_{n,i} + \Delta\mathbf{W}_i)\mathbf{y}_i.\end{aligned} \tag{9}$$

Since each term appearing in Eq. (9) presents the same structure, we develop the linearization for one of these terms:

$$\mathbf{y}_j^{\mathsf{T}}(\mathbf{W}_{n,j}^{\mathsf{T}} + \Delta\mathbf{W}_j^{\mathsf{T}})(\mathbf{W}_{n,i} + \Delta\mathbf{W}_i)\mathbf{y}_i \approx \mathbf{y}_j^{\mathsf{T}}\mathbf{W}_{n,j}^{\mathsf{T}}\mathbf{W}_{n,i}\mathbf{y}_i + \mathbf{y}_j^{\mathsf{T}}\mathbf{W}_{n,j}^{\mathsf{T}}\Delta\mathbf{W}_i\mathbf{y}_i + \mathbf{y}_i^{\mathsf{T}}\mathbf{W}_{n,i}^{\mathsf{T}}\Delta\mathbf{W}_j\mathbf{y}_j, \tag{10}$$

and similarly for the other terms in Eq. (9).

The nonlinear model is approximated according to

$$\Delta \mathbf{W}_i = \sum_{k=1}^{\mathrm{K}} \Delta \mathbb{W}_k \mathcal{P}_k(\mathbf{y}_i). \tag{11}$$

Eq. (8) concerns data $\mathbf{y}_i$ and $\mathbf{y}_j$. Hence, by applying the same procedure for any data pair, an algebraic system of equations results involving the $N \times d \times K$ unknowns. Thus, the safe construction of the non-linear mapping requires a rich enough sampling, i.e.

$$N \times d \times K < \frac{P^2}{2} - P \tag{12}$$

The nonlinear approximation (11) was accomplished by using radial basis based on either the gaussian or the inverse-quadratic radial kernels, both making use of the distance r

$$r_k = ||\mathbf{y} - \mathbf{y}_k||, \tag{13}$$

from which both interpolants read

$$\mathcal{P}_k(r) = \mathrm{e}^{-(\epsilon r_k)^2}, \tag{14}$$

and

$$\mathcal{P}_k(r) = \frac{1}{\sqrt{1 + \mathrm{e}^{-(\epsilon r_k)^2}}}, \tag{15}$$

with ϵ affecting the shape of the radial kernel.

References

[1] S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[2] E. Lopez, D. Gonzalez, J.V. Aguado, E. Abisset-Chavanne, E. Cueto, C. Binetruy, F. Chinesta, Archives of computational methods in engineering, Int. J. Numer. Methods Eng. 25 (1) (January 2018) 59–68, https://doi.org/10.1007/s11831-016-9172-5.

[3] L. Vans Der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.

[4] A. Criminisi, J. Shotton, E. Konukoglu, Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Microsoft Research technical report, TR-2011-114, 2011.

[5] J. Greenhalgh, M. Miermehdi, Traffic sign recognition using Mser and Random forests, in: 20th European Signal Processing Conference, 2012.

[6] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Netw. 61 (2015) 85–117.

[7] L. Yann, B. Yoshua, H. Geoffrey, Deep learning, Nature 521 (2015) 436–444.

[8] J. Donahue, L.A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2017) 677–691.

[9] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, IEEE Trans. Pattern Anal. Mach. Intell. 38 (2) (2016) 295–307.

[10] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[11] P. Norvig, S. Russell, Artificial Intelligence, a Modern Approach, Prentice-Hall, 1994.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representation of words and phrases their compositionality, in: Proceeding NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, 2013.

[13] M. Abadi, et al., Google Research Large-Scale Machine Learning on Heterogeneous Distributed Systems, Technical Report, 2015.

[14] R. Ibanez, E. Abbisset-Chavanne, A. Ammar, D. Gonzalze, E. Cueto, J.-L. Duval, F. Chinesta, A multidimensional data-driven sparse identification technique: the sparse proper generalized decomposition, Complexity (2018) 5608286, https://doi.org/10.1155/2018/5608286.

[15] G. Quaranta, C. Argerich, R. Ibanez, J.-L. Duval, E. Cueto, F. Chinesta, From linear to nonlinear PGD-based parametric structural dynamics, C. R. Mécanique 347 (2019) 445–454.