Efficient methodology for seismic fragility curves estimation by Active Learning on Support Vector Machines

Rémi Sainct*, Cyril Feau*, Jean-Marc Martinez, Josselin Garnier

*Den-Service d’études mécaniques et thermiques (SEMT), CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France

Den-Service de thermo-hydraulique et de mécanique des fluides (STMF), CEA, Université Paris-Saclay, F-91191 Gif-sur-Yvette, France

Ecole Polytechnique, CMAP, 91128 Palaiseau Cedex, France

Abstract

Fragility curves which express the failure probability of a structure, or critical components, as function of a loading intensity measure are nowadays widely used (i) in Seismic Probabilistic Risk Assessment studies, (ii) to evaluate impact of construction details on the structural performance of installations under seismic excitations or under other loading sources such as wind. To avoid the use of parametric models such as lognormal model to estimate fragility curves from a reduced number of numerical calculations, a methodology based on Support Vector Machines coupled with an active learning algorithm is proposed in this paper. In practice, input excitation is reduced to some relevant parameters and, given these parameters, SVMs are used for a binary classification of the structural responses relative to a limit threshold of exceedance. Since the output is not binary but a real-valued score, a probabilistic interpretation of the output is exploited to estimate very efficiently fragility curves as score functions or as functions of classical seismic intensity measures.

Keywords: Fragility curve, Active Learning, Support Vector Machines, seismic intensity measure indicator

*Corresponding author

Email addresses: remi.sainct@m4x.org (Rémi Sainct), cyril.feau@cea.fr (Cyril Feau), jean-marc.martinez@cea.fr (Jean-Marc Martinez), josselin.garnier@polytechnique.edu (Josselin Garnier)
1. Introduction

In Seismic Probabilistic Risk Assessment (SPRA) studies performed on industrial facilities, a key point is the evaluation of fragility curves which express the failure probability of a structure, or critical components, as a function of a seismic Intensity Measure (IM) such as the peak ground acceleration (PGA) or the spectral acceleration. It should be noted that apart from the use in a SPRA framework (see e.g. [1, 2, 3, 4, 5]), fragility curves can be used for making decisions regarding the choice of construction details, to improve the structural performance of installations under seismic excitations (see e.g. [6, 7, 8, 9]) or hurricanes (see e.g. [10]). They can also be used to evaluate impact of ground motion characteristics (near-fault type like, broadband, see e.g. [9, 11]), impact of soil-structure interaction (see e.g. [12]) or impact of numerical modeling assumptions (see e.g. [13]) on the seismic vulnerability assessment, etc. Finally, it is worth noting that the use of fragility curves is not limited to seismic excitation, they can also be applied to other loading sources such as wind (see e.g. [14, 10]).

In theory, for complex structures, fragility curves have to be evaluated empirically based on a large number of mechanical analyses requiring, in most cases, nonlinear time-history calculations including both the uncertainties inherent to the system capacity and to the seismic demand, respectively called epistemic and aleatory uncertainties [1, 3, 15]. Nevertheless, the prohibitive computational cost induced by most of nonlinear mechanical models requires the development of numerically efficient methods to evaluate such curves from a minimal number of computations, in particular in industrial contexts.

Following the idea proposed in the early 1980’s in the framework of nuclear safety assessment [1], the lognormal parametric model has been widely used in many applications to estimate fragility curves from a reduced number of numerical calculations (see e.g. [2, 3, 4, 5, 6, 7, 8, 11, 12, 16, 17]). Different methods can be employed to determine the parameters of the lognormal model (see e.g. [2, 4, 18, 19]) and different model assumptions can also be considered (see e.g. [16, 17, 20]). However, the validity of the parametric models is itself questionable. For example, considering a steel frame structure in [19], the authors show that the accuracy of the lognormal curves depends on the ground motion IM, the failure criterion and the employed
method for fitting the model. This last result is also obtained in [16] for a typical Indian PHWR containment. In [17], the influence of large data points in the fit results is highlighted in a concrete-dams case study. In [20], a general framework for the estimation of analytical fragility functions based on bivariate distributions which includes two criteria for parametric model selection is first proposed. Then, the author pointed out the influence of the model assumptions in the fragility curve estimation of a reinforced concrete building.

The question of the representativity is inevitable with the use of parametric models since, for the complex cases of interest, it is very difficult to verify their validity. So, to bypass this problem, the need of a numerically efficient non parametric-based methodology (which would be accurate with a minimum number of mechanical analyses) is necessary. A way to achieve this goal consists in building a metamodel (i.e. a surrogate model of the mechanical analysis) which expresses the statistical relation between seismic inputs (seismic IM indicators) and structural outputs also called Engineering Demand Parameters (EDP). Various metamodeling strategies have been proposed recently in the literature based on, for example, response surfaces ([21, 22]), kriging [9] and Artificial Neural Networks (ANNs) [23]. In [10], considering storage tanks subjected to hurricane induced storm surge, prediction accuracy of three metamodels (Support Vector Machines, Random Forest and Logistic Regression) has been systematically assessed and compared for various failure modes and Logistic Regression models were found to be the most accurate in order to estimate fragility curves.

The goal of this paper is twofold. First, it is to propose a simple and efficient methodology for estimating non parametric fragility curves that allows to reduce the number of mechanical numerical simulations by optimizing their selection. Second, it is to address the question of the best seismic IM indicator that can to be used as abscissa of the fragility curves and not be limited to the PGA or the spectral acceleration. To this end, the strategy proposed is based on the use of Support Vector Machines (SVMs) coupled with an active learning algorithm. In our case, SVMs are used for a binary classification of structural responses relative to a limit threshold of exceedance. By this way, the two problems can be addressed together. Indeed, since the SVM output is not binary but a real-valued score (SVM’s margin), a probabilistic interpretation of this real-valued score can be introduced to estimate very efficiently fragility curves. Regarding the question of the IM indicators, a complete review of those proposed in the literature can be found in [17] as well as five
optimality criteria (efficiency, practicality, proficiency, sufficiency and hazard compatibility). According to these criteria, the methodology proposed here consists in defining a proficient IM, proficiency being a composite measure of efficiency and practicality. Indeed, the score function can be viewed as an efficient seismic IM since a perfect classifier would lead to a fragility curve in the form of a unit step function when the problem is linearly separable. Moreover, as we will see, as the score function will be highly correlated with the EDP (in the logarithmic scale), it can be considered as a practical IM. Hazard compatibility and sufficiency are considered as out of the scope of this paper and will be addressed in future work.

In contrast to classical learning (passive learning), the active learner selects the most useful numerical experiments to be carried out and added to the learning data set. The "learners" choose the best instances from a given large set of unlabeled examples. So, the main question in active learning is how to choose new numerical experiments to be labeled. Various methods proposed in active learning by ANNs are presented in [24]. Most are based on the learning of several "learners" ([25, 26]). With SVMs, active learning can be done very easily by using only one learner because the distance to the separator hyperplane is a "natural" criterion for selecting new points to "label" [27]. A similar technique using logistic output neural networks can be used by analyzing the logit of the output. But in this case, given the non-linearity of the ANNs, the different learnings of the learner may present a strong variability on the decision boundary.

The recent progress on the simulation of seismic ground motions makes it possible to propose a methodology based on an active learning approach, which requires a number of realizations larger than the size of the available real signals databases (in order to optimize their selection), but small enough to be able to use complex mechanical models. Various techniques can be used to create artificial seismic signals (see e.g. the review presented in [28] and non-exhaustive references [29], [30], [31], [32]). In this work, we have chosen to enrich a set of acceleration records selected in a real ground motion database using magnitude and distance criteria. To this end, the parameterized stochastic model of modulated and filtered white-noise process defined in [33] has been implemented. This model efficiently addresses both temporal and spectral nonstationarities of seismic signals and has been used in several recent works such as [19], [34], [35], [36] and [37]. The advantage of this kind of model is that its constitutive parameters that characterize its time-frequency envelope can be considered as input parameters of a metamodel.
additionally with the classical IM parameters (PGA, spectral acceleration, etc.), although it is a non-traditional approach. Indeed, although the links between the ground motion simulation model’s parameters and the nonlinear structural responses are not necessarily trivial, intrinsically these parameters contain information that can help to discriminate seismic signals from the point of view of their damaging potential. A machine learning-based approach allows to capture such links which are not intuitive.

The methodology proposed in this paper to estimate fragility curves consists first of all in generating a large set of artificial seismic signals and to compute the different IM indicators of interest. In practice, this step is not time consuming in contrast with the nonlinear mechanical calculations. Then, the second step consists in building a SVM-based classifier by optimally selecting by active learning the mechanical calculations to perform. A probabilistic interpretation of the real-valued score given by the classifier is used in a third step to estimate very efficiently fragility curves as score functions. However, the classifier can also be used to predict the scores and probabilities associated to several new input parameters (those which have not been selected for the definition of the classifier or others generated from new simulations of the ground motion model) in order to estimate fragility curves as functions of the classical seismic IMs. Different procedures can be used to construct empirical fragility curves (see e.g. [38], [19]). Here, we assume that they are simply evaluated based on k-means clustering of the IM data. In a Monte Carlo-based approach this means that in each cluster, the empirical probability of failure corresponds to the ratio between the number of structural responses that exceed the limit threshold and the cluster size, that is, the number of structural responses belonging to the cluster.

In this paper, a brief summary of the ground motion simulation model that has been implemented for this study is presented in section 2. In order to validate the methodology at a large scale within a Monte Carlo-based approach that does not require any assumption to estimate probabilities of interest, a simple inelastic oscillator has been considered. This structural model is also presented in section 2 as well as the IM indicators that have been chosen. Section 3 is devoted to the presentation of the different classification methods, and the active learning methodology. Section 4 explains how the proposed methodology makes it possible to estimate fragility curves, using either the score functions or the classical IM indicators. Finally, a conclusion is presented in section 5.
2. Ground motion simulation model, model of the mechanical structure and choice of the IM indicators

In this section, a brief summary of the Ground Motion Simulation Model (GMSM) that has been implemented for this work is first presented. Then, the model of the mechanical structure which is used to illustrate the methodology is presented as well as the choice of the IM indicators selected as inputs of the SVMs. Finally, a discussion is proposed regarding the GMSMs.

2.1. Model of earthquake ground motion

Following Rezaeian and Der Kiureghian [39], a seismic ground motion \( s(t) \) with \( t \in [0, T] \) is modeled as:

\[
s(t) = q(t, \alpha) \left[ \frac{1}{\sigma_f(t)} \int_{-\infty}^{t} h[t - \tau, \lambda(\tau)] w(\tau) d\tau \right],
\]

where \( q(t, \alpha) \) is a deterministic, non-negative modulating function with a set of parameters \( \alpha \), and the process inside the squared brackets is a filtered white-noise process of unit variance: \( w(t) \) is a white-noise process, \( h(t, \lambda) \) denotes the impulse response function (IRF) of the linear filter with a set of parameters \( \lambda \), and \( \sigma_f(t) = \sqrt{\int_{-\infty}^{t} h^2(t - \tau, \lambda(\tau)) d\tau} \) is the standard deviation of the process defined by the integral in equation 1.

In order to achieve spectral nonstationarity of the ground motion, the parameters \( \lambda \) of the filter depend on the time \( \tau \) of application of the pulse; thus the standard deviation \( \sigma \) depend on \( t \). Still following Rezaeian, we choose for the impulse response function:

\[
h[t - \tau, \lambda(\tau)] = \frac{\omega_f(\tau)}{\sqrt{1 - \zeta_f^2}} \exp \left[ -\zeta_f \omega_f(\tau)(t - \tau) \right] \sin \left[ \omega_f(\tau) \sqrt{1 - \zeta_f^2} (t - \tau) \right] \text{ if } t \geq \tau,
\]

\[
= 0 \text{ otherwise},
\]

where \( \lambda(\tau) = [\omega_f(\tau), \zeta_f] \) is the set of parameters, \( \omega_f(\tau) \) is the natural frequency (dependent on the time of application of the pulse) and \( \zeta_f \in [0, 1] \) is the (constant) damping ratio. A linear form is chosen for the frequency: \( \omega_f(\tau) = \omega_0 + \frac{\tau}{T} (\omega_n - \omega_0) \). The modulating function \( q(t, \alpha) \) is defined piece-wise:
\[ q(t, \alpha) = \begin{cases} 0 & \text{if } t \leq T_0, \\ \alpha_1 \left( \frac{t-T_0}{T_1} \right)^2 & \text{if } T_0 \leq t \leq T_0 + T_1, \\ \alpha_1 & \text{if } T_0 + T_1 \leq t \leq T_0 + T_2, \\ \alpha_1 \exp \left[ -\alpha_2 (t - T_0 - T_2)^{\alpha_3} \right] & \text{if } t \geq T_0 + T_2. \end{cases} \] (3)

The modulation parameters are thus: \( \alpha = (\alpha_1, \alpha_2, \alpha_3, T_0, T_1, T_2) \). The initial delay \( T_0 \) is used in parameter identification from real ground motions, but it is not used in simulations (we choose \( T_0 = 0 \)). To summarize, the generated signals are associated with 8 real parameters: \( (\alpha_1, \alpha_2, \alpha_3, T_1, T_2, \omega_0, \omega_n, \zeta_f) \).

Finally, a high-pass filter is used as post-processing to guarantee zero residuals in the acceleration, velocity and displacement. The corrected signal \( \ddot{u}(t) \) is the solution of the differential equation:

\[ \ddot{u}(t) + 2 \omega_c \dot{u}(t) + \omega_c^2 u(t) = s(t), \] (4)

where \( \omega_c = 0.2 \) Hz is the corner frequency. Due to high damping of the oscillator (damping ratio of 100\%), it is clear that \( u(t), \dot{u}(t) \) and \( \ddot{u}(t) \) all vanish shortly after the input process \( s(t) \) has vanished, thus assuring zero residuals for the simulated ground motion. In the rest of this paper we will call \( s(t) \) the corrected signal \( \ddot{u}(t) \).

The identification of the model parameters \( \theta = (\alpha, \lambda) \) for each of the \( N_r = 97 \) acceleration records, gives \( N_r \) data points \( (\theta_i)_{i=1}^{N_r} \) in the parameter space \( (\mathbb{R}^8 \text{ in this case}) \). The model then allows to generate any number of artificial signals, thanks to the white noise \( w(t) \). However, these signals would all have very similar features due to the limited number of real signals considered to define the GMSM. In order to estimate a fragility curve, we need to be able to generate artificial signals over a whole range of magnitudes, with realistic associated probabilities. Thus, we have to add a second level of randomness in the generation process, coming from the parameters themselves. To do this, the parameters’ distribution has been estimated using a Gaussian Kernel.
Density Estimation (KDE) with a multivariate bandwidth estimation, as defined by Kristan in [41]. Thus, the estimator reads:

\[ p_{KDE}(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \phi_H(\theta - \theta_i), \]  

(5)

\( \phi_H \) is a Gaussian kernel centered at 0 with covariance matrix \( H \) properly chosen from the data points \( \{\theta_i\}_{i=1}^{N_r} \). The complete definition of the estimator \( p_{KDE}(\theta) \) is presented in Appendix A.3 for completeness.

Finally, the simulation of an artificial ground motion requires three steps:

- choose an integer \( i \in [1, N_r] \) with a uniform distribution;
- sample a vector \( y \) from a Gaussian distribution with probability density function \( \phi_H \) centered at 0 with covariance matrix \( H \), and let \( \theta = \theta_i + y \);
- sample a white noise \( w(\tau) \) and compute the signal using (1), with parameters \( (\alpha, \lambda) = \theta \).

For this work we generated \( N_s = 10^5 \) artificial seismic ground motions \( s_i(t) \) using this method.

2.2. Model of the mechanical structure

For the illustrative application of the methodology developed in this paper, a nonlinear single degree of freedom system is considered. Indeed, despite its extreme simplicity, such model may reflect the essential features of the nonlinear responses of some real structures. Moreover, in a probabilistic context requiring Monte Carlo simulations, it makes it possible to have reference results with reasonable numerical cost. Its equation of motion reads:

\[ \ddot{z}_i(t) + 2\beta \omega_L \dot{z}_i(t) + f_{nl}^i(t) = -s_i(t), \quad i \in [1, N_s] \]  

(6)

where \( \dot{z}_i(t) \) and \( \ddot{z}_i(t) \) are respectively the relative velocity and acceleration of the unit mass of the system submitted to the \( i \)th artificial seismic ground motion \( s_i(t) \) with null initial conditions in velocity and displacement. In equation 6, \( \beta \) is the damping ratio, \( \omega_L = 2\pi f_L \) is the circular frequency and \( f_{nl}^i(t) \) is the nonlinear resisting force.

The rheological hysteretic model of the oscillator, defining kinematic hardening, is presented in Figure 1. In this study, \( f_L = 5 \) Hz, \( \beta = 2\% \),
the yield limit is \( Y = 5.10^{-3} \text{ m} \), and the post-yield stiffness \( am\omega^2_L \) (see Figure 1) is equal to 20% of the elastic stiffness \( m\omega^2_L \), that is \( a = 0.2 \).

Moreover, we call \( \tilde{z}_i(t) \) the relative displacement of the associated linear system \( (a = 1 \text{ see Figure 1}) \), that is assumed to be known in the sequel, whose equation of motion is:

\[
\ddot{\tilde{z}}_i(t) + 2\beta\omega_L \dot{\tilde{z}}_i(t) + \omega^2_L \tilde{z}_i(t) = -s_i(t),
\]

and we set:

\[
Z_i = \max_{t \in [0,T]} |z_i(t)|,
\]

\[
L_i = \max_{t \in [0,T]} |\tilde{z}_i(t)|.
\]

In this work, equations 6 and 7 are solved numerically with a finite-difference method.

### 2.3. Choice of the seismic IM indicators

A complete review of the existing seismic IM indicators in the literature can be found in [17]. In a complementary way, the methodology proposed here is intended to take into account the advantage of using a ground motion parametric model like that of Rezaeian and Der Kiureghian in considering its constitutive parameters as input parameters of a metamodel. Thus, if \( \mathcal{B} = (s_i(t))_{i \in [1,N_s]} \) is the database of \( N_s \) simulated ground motions, we can consider \( \theta_i = (\alpha_i, \lambda_i) \in \mathbb{R}^8 \) the associated modulating and filter parameters as inputs. However, they can not be used alone since there is an infinity of possible realizations of the stochastic process for a set of parameters, due to the white-noise process. They have to be used additionally with the main classical IM parameters. Thus, for every signal \( s_i(t) \), we also consider:
• the peak ground acceleration

\[ PGA_i = \max_{t \in [0, T]} |s_i(t)|; \]

• the maximum velocity (or Peak Ground Velocity)

\[ V_i = \max_{t \in [0, T]} \left| \int_0^t s_i(\tau) d\tau \right|; \]

• the maximum displacement (or Peak Ground Displacement)

\[ D_i = \max_{t \in [0, T]} \left| \int_0^t \int_0^\tau s_i(u) du d\tau \right|; \]

• the total energy \( E_i = E_{s_i}(T) = \int_0^T s_i^2(\tau) d\tau \) (this IM parameter is proportional to the "Arias Intensity" indicator usually considered);

• the maximum linear displacement \( L_i \) of the associated linear structure (equation 9). The spectral acceleration \( (\omega_L^2 L_i) \) is usually considered as IM indicator. Nevertheless, since here the variable of interest is a non-linear displacement, it is more suitable to use spectral displacement. This structure-dependent indicator is used in order to improve the performance of the metamodeling strategy since it is highly correlated with the output that is, the maximum displacement of the nonlinear structural response (equation 8).

It is worth noting that the use of the classical parameters in association with the ground motion’s ones does not guarantee that it will be possible to find a perfect "meta-indicator", i.e. a perfect combination of the input indicators that can predict the failure of the structure with a very high accuracy. In fact, we know that this is not possible, because for a given value of PGA (or more exactly a given value of PGA belonging to a "small interval"), there is an infinite number of possible realizations of the ground motion. This also applies for the other IM indicators considered and this is the main challenge of this kind of problem. However, we will see in the paper that the proposed methodology allows to estimate very efficiently the fragility curves of interest with a very good precision. So, for each simulated signal we have a vector \( X_i^* = (\alpha_i, \lambda_i, PGA_i, V_i, D_i, E_i, L_i) \in \mathbb{R}^{13} \) of 13 real parameters. We want to predict whether the maximum total displacement \( Z_i \) of the nonlinear structure is greater than a certain threshold, for example twice the elasticity limit \( Y \).
2.4. Representativity of the GMSM

In order to assess the representativity of the GMSM, using the linear equation 7, we can compare the response spectra of the $N_r = 97$ recorded accelerograms with that of the $N_s = 10^5$ simulated signals. Figure 2a shows this comparison for the average spectrum, as well as the 0.15, 0.5 and 0.85 quantiles. It can be seen that the simulated signals have statistically the same response spectra as the real signals, although at high frequency ($f_L > 30$ Hz), the strongest simulated signals have higher responses than the real ones. This may be due to the fact that the model conserves energy (section Appendix A.1), while the selection of acceleration records from ESMD is based on magnitude. To illustrate this, figures 2b and 2c show the empirical cumulative distribution functions of the PGA and total energy. While there is a good match for the energy, the PGA of the strongest simulated signals is slightly higher than for the real ones.
Figure 2: Comparison between the real and simulated data bases. (a) Response spectra for 2% damping ratio. Zoom on the empirical cumulative distribution functions of (b) the PGA and (c) total energy.

2.5. Discussion

As mentioned in the introduction, in practice different GMSMs can be used to enrich a database of real seismic signals, knowing that there is really no consensus regarding the one (resp. those) which has (resp. have) to be privileged in a study (see e.g. the references [28], [29], [30], [31] and [32]). To the best of the knowledge of the authors, if a generator is defined from real seismic signals, it will be considered as valid if the statistical characteristics of the artificial signals are close to the ones of the real signals reduced to some scalar indicators (e.g. $PGA_i$, $E_i$ see Figures 2b and 2c, and non-exhaustively $V_i$, $D_i$ not shown here for space reason), starting with the response spectra (see Figure 2a). Another requirement is that the realizations of the GMSM
should be as decorrelated as possible to lead to enough variability in the structural responses. All conditions are met here with the Rezaeian and Der Kiureghian’s model and especially the last one because of the presence of the white-noise process. This is not necessarily the case of all the models, as the ones based on the Karhunen–Loève expansion for example (see e.g. reference [32]) whereas they give good adequation with the statistical indicators related to real signals. With this kind of models, the uncorrelated random variables of the linear combination of the orthogonal basis functions, could be considered as the input parameters of the metamodel. Nevertheless, if the number of orthogonal basis functions used in the decomposition is not large enough (e.g. due to the paucity of the real signals used to define the model), it can lead to artificial signals that have too high levels of correlation. Consequently, in practice, the possibility to include the GMSM’s parameters, as well as their nature, depends on the type of model considered. The accuracy of the metamodel predicting the maximum total displacement may also depend on the type of GMSM chosen if the parameters are more or less well correlated with the output of interest. However, the influence of the GMSM is out of the scope of this work which aims to propose a general methodological framework to find an ”optimal” classifier (or indicator) for a non-prohibitive computational cost, whatever the structure and the GMSM considered.

3. Binary Classification and Active Learning

In this section, a simple but crucial preprocessing of the data is first addressed. Indeed, although the results of the comparative study are not shown here for the sake of brevity, this preprocessing improves the performance of the classifier. Then, different classification methods and the active learning methodology are presented. The active learning methodology consists in selecting sequentially $n$ signals $s_i(t)$ and computing the corresponding maximum total displacements $Z_i$ (which is the time-consuming step) in order to build a classifier that will predict whether the maximum total displacement of a new signal $s(t)$ exceeds the threshold $2Y$. Finally, two tests of performance are presented. These tests are based on a large set of $N$ total displacements $Z$, with $N \gg n$. So, considering the simple inelastic structure presented in section 2.2, which allows calculations of $N$ structural responses with a reasonable computational time, the performances of different classifiers can be compared. The results of this comparison are then used to formulate reom-
mendations on the values of $n$ that can be used in practice to estimate the
fragility curves of complex structures (see section 3.4).

3.1. Preprocessing of the training data

First of all, recall that we performed $N_s = 10^5$ simulations of seismic
signals and for each of them, we calculate the displacement $L_i$ of the associ-
ated linear mechanical structure. This step is not time consuming in contrast
with the nonlinear mechanical calculations. Next, signals with very small and
large values of $L$ are discarded from the database. Indeed, the signals whose
maximum linear displacements are less than the elasticity limit $Y$ are not
interesting, since we know they do not reach the limit threshold of interest
$(2Y)$:

$$L_i < Y \Rightarrow Z_i = L_i \quad \text{and thus} \quad Z_i < Y.$$  

This discards 66% of the simulated signals. We also discard a few signals
(0.3%) whose maximum linear displacements are too large ($L_i > 6Y$), since
the mechanical model we use is not realistic beyond that level. This gives a
subset $I$ of our database such that:

$$\forall i \in I \quad L_i \in [Y, 6Y].$$

This leads to a size $N = 33718$ for the data set. On those $N$ signals, a
Box-Cox transform is applied to the thirteen entries of $X_i^* = (\alpha_i, \lambda_i, PGA_i, V_i, D_i, E_i, L_i) \in \mathbb{R}^{13}$. This nonlinear step is critical for the accuracy of the
classification, especially when we later use linear SVM classifiers. The Box-
Cox transform is defined by:

$$BC_\delta(x) = \begin{cases} 
\frac{x^\delta - 1}{\delta} & \text{if } \delta \neq 0 \\
\frac{\log(x)}{\delta} & \text{if } \delta = 0.
\end{cases} (10)$$

The parameter $\delta$ is optimized, for each entry, in order to obtain a statisti-
cal distribution close to the normal law by maximizing the log-likelihood. Fig-
ure 3 shows that the thirteen entry $L$, i.e. the linear displacement, is heavily
modified by this transformation, with an optimal parameter of $\delta = -0.928$.

Finally, all of the 13 components are standardized, thus forming the train-
ing database $X = \{X_1, \ldots, X_N\}$ with $X_i \in \mathbb{R}^{13}$. 

14
3.2. SVMs, Active Learning and performance checking

3.2.1. Simple classifiers

At the most basic level, a binary classifier is a labeling function

\[ \hat{l} : \mathbb{R}^d \rightarrow \{-1; 1\} \]

\[ \mathbf{X} \mapsto \hat{l}(\mathbf{X}), \]

that, given a vector \( \mathbf{X} \in \mathbb{R}^d \) corresponding to a seismic signal \( s(t) \), gives us an estimated label \( \hat{l} \). In our setting, the true label \( l_i \) of instance \( \mathbf{X}_i \) is 1 if the displacement \( Z_i \) is greater than the damage threshold \( 2Y \), and \(-1\) otherwise:

\[ l = sgn(Z - 2Y) = \begin{cases} 1 & \text{if } Z > 2Y, \\ -1 & \text{otherwise.} \end{cases} \]  

Note that the true label \( l_i \) is not in general a function of the vector \( \mathbf{X}_i \), since it depends on the full signal \( s_i(t) \) when \( \mathbf{X}_i \) only gives us macroscopic measures of the signal; therefore, a perfect classifier \( \hat{l}(\mathbf{X}_i) \) may not exist (see section 2.3).

One of the simplest choice for a classifier is to look at only one component of the vector \( \mathbf{X} \). For example, it is obvious that the PGA is highly correlated with the maximum total displacement \( Z \); therefore, we can define the PGA classifier \( \hat{l}_{PGA} \) as:

\[ \hat{l}_{PGA}(\mathbf{X}) = sgn(PGA - M) \]
where \( M \) is a threshold to be adjusted. Moving the threshold up results in less false positives (\( \hat{l} = 1 \) when the real label is \( l = -1 \)) but more false negatives (\( \hat{l} = -1 \) when the real label is \( l = 1 \)); and moving the threshold down results in the opposite. Therefore, there exists a choice of \( M \) such that the number of false positives and false negatives are equal, as can be seen in figure 4.

![Figure 4: Choice of the threshold for the binary PGA classifier \( \hat{l}_{PGA} \).](image)

Note that this choice does not guarantee that the total number of misclassifications is minimal. Similarly, we can also define a classifier \( \hat{l}_L \) based on the maximum linear displacement \( L \), since the linear displacement is also highly correlated with the total displacement. These two simple classifiers give us a baseline to measure the performance of more advanced classifiers.

### 3.2.2. Support Vector Machines

In machine learning, support vector machines (SVMs) are supervised learning models used for classification and regression analysis. In the linear binary classification setting, given a training data set \( \{X_1, \ldots, X_n\} \) that are vectors in \( \mathbb{R}^d \), and their labels \( \{l_1, \ldots, l_n\} \) in \( \{-1, 1\} \), the SVM is a hyperplane of \( \mathbb{R}^d \) that separates the data by a maximal margin. More generally, SVMs allow one to project the original training data set \( \{X_1, \ldots, X_n\} \) onto a higher dimensional feature space via a Mercer kernel operator \( K \). The classifier then associates to each new signal \( X \) a score \( f_n(X) \) given by:

\[
f_n(X) = \sum_{i=1}^{n} \varphi_i K(X_i, X).
\] (14)

A new seismic signal represented by the vector \( X \) has an estimated label \( \hat{l} \) of 1 if \( f_n(X) > 0 \), \(-1\) otherwise. In a general SVM setting, most of the labeled instances \( X_i \) have an associated coefficient \( \varphi_i \) equal to 0; the few vectors \( X_i \) such that \( \varphi_i \neq 0 \) are called "support vectors", thus the name "support vector machine". This historical distinction among labeled instances is less relevant in the case of active learning (see next section), since most of the \( \varphi_i \) are
non-zero. In the linear case, \( K(X_i, X) \) is just the scalar product in \( \mathbb{R}^d \), and the score is:

\[
    f_n(X) = W^T X + c,
\]

where \( W \in \mathbb{R}^d \) and \( c \in \mathbb{R} \) depend on the coefficients \( \varphi_i \). Another commonly used kernel is the radial basis function kernel (or RBF kernel) \( K(U, U') = e^{-\gamma(U-U')(U-U')} \), which induces boundaries by placing weighted Gaussians upon key training instances.

3.2.3. Active Learning: basic principles

Computing the total displacement \( Z_i \) of the structure (and thus the label \( l_i \)) is very costly for a complex structure, limiting the size of the training data. Fortunately, it is possible to make accurate classifiers using only a limited number of labeled training instances, using active learning.

In the case of pool-based active learning, we have, in addition to the labeled set \( \mathcal{L} = \{X_1, \ldots , X_n\} \), access to a set of unlabeled samples \( \mathcal{U} = \{X_{n+1}, \ldots , X_N\} \) (therefore we have \( \mathcal{X} = \mathcal{L} \cup \mathcal{U} \)). We assume that there exists a way to provide us with a label for any sample \( X_i \) from this set (in our case, running a full simulation of the physical model using signal \( s_i(t) \)), but the labeling cost is high. After labeling a sample, we simply add it to our training set. In order to improve a classifier it seems intuitive to query labels for samples that cannot be easily classified. Various querying methods are possible [42, 27], but the method we present here only requires to compute the score \( f_n(X) \) for all samples in the unlabeled set, then to identify a sample that reaches the minimum of the absolute value \( |f_n(X)| \), since a score close to 0 means a high uncertainty for this sample. Thus, we start with \( n = 2 \) samples with indices \( j_1 \) and \( j_2 \), labeled +1 and −1. Recursively, if we know the labels of signals with indices \( j_1, \ldots , j_n \):

- we compute the SVM classifier associated with the labeled set \( \{(X_{j_1}, l_{j_1}), \ldots , (X_{j_n}, l_{j_n})\} \);
- for each unlabeled instance \( X_i, i \in [1, N] \setminus \{j_1, \ldots , j_n\} \), we compute its score \( f_n(X_i) \);
- we determine the instance with maximum uncertainty for this classifier:

\[
    j_{n+1} = \arg \min_{i \in [1, N] \setminus \{j_1, \ldots , j_n\}} |f_n(X_i)|,
\]
and compute the corresponding maximum total displacement $Z_{j_{n+1}}$ by running a full simulation of the physical model;

- the instance $(X_{j_{n+1}}, l_{j_{n+1}} = sgn(Z_{j_{n+1}} - 2Y))$ is added to the labeled set.

No termination criteria are explicitly presented here since, in practice, the limitation regarding the number of training data available is mainly due to the computational cost of the numerical mechanical calculations. Nevertheless, the extensive simulations we have done allow us to make recommendations on the minimum number of simulations to be carried out (see section 3.4).

3.2.4. Active Learning : choice of the starting points

The active learner needs two starting points, one on each side of the threshold. After the preprocessing step, about 17% of the $N$ remaining instances have a displacement greater than the threshold (although this precise value is usually unknown). It can be tempting to choose, for example, the signal with the smallest PGA as $j_1$ and the signal with the biggest PGA as $j_2$. However, running simulations with these signals is costly and give us a relatively useless information. We prefer to choose the starting points randomly, which also allows us to see how this randomness affects the final performance of the classifier.

The linear displacement $L_i$ and the PGA of a signal are both obviously strongly correlated with the displacement $Z_i$. As a consequence, it is preferable that the starting points respect the order for these two variables:

$$Z_{j_1} < 2Y < Z_{j_2}, \quad L_{j_1} < L_{j_2} \quad \text{and} \quad PGA_{j_1} < PGA_{j_2}. \quad (17)$$

Indeed, if $j_1$ and $j_2$ are such that, for example, $Z_{j_1} < 2Y < Z_{j_2}$ but $PGA_{j_1} > PGA_{j_2}$, then the active learner starts by assuming that the PGA and displacement have a negative correlation, and it can take many simulations before it “flips”; in some rare instances the classifier performs extremely poorly for several hundreds of simulations. Thus, the starting points $j_1$ and $j_2$ are chosen such that equation (17) is automatically true, using quantiles of the PGA and linear displacement. $j_1$ is chosen randomly among the instances whose PGA is smaller than the median PGA and whose linear displacement is smaller than the median linear displacement:
\[ j_1 \in \{i \in [1, N] \mid PGA_i < D_5(PGA) \quad \& \quad L_i < D_5(L) \}, \quad (18) \]

where \( D_5(PGA) \) denotes the median of set \( PGA_1, ..., PGA_N \), and similarly for \( D_5(L) \). It is almost certain that any instance in this set satisfies \( Z_i < 2Y \) and thus \( l_i = -1 \). Similarly, \( j_2 \) is chosen using the 9th decile of both PGA and linear displacement:

\[ j_2 \in \{i \in [1, N] \mid PGA_i > D_9(PGA) \quad \& \quad L_i > D_9(L) \}, \quad (19) \]

where \( D_9(PGA) \) denotes the 9th decile of set \( PGA_1, ..., PGA_N \), and similarly for \( D_9(L) \). The probability that \( Z_i > 2Y \) in this case was found to be 97%. If we get unlucky and \( Z_i < 2Y \) then we discard this signal and choose another one.

3.2.5. Performance checking: ROC curve and precision/recall breakeven point

Two tests of performance are proposed in this section. They are based on a large set of \( N \) total displacements \( Z \), with \( N \gg n \) and \( n \) the number of instances required by the active learning algorithm to train the classifier. Recall that since it is practically impossible to obtain such a large set for complex structures, our method is applied here on the simple inelastic structure presented in section 2.2 in order to be able to check the results and make recommendations on the values of \( n \) that can be used in practice.

The SVM classifier gives an estimated label \( \hat{l}_i \) to each signal \( s_i \) depending on its score \( \hat{l}_i = sgn(f_n(X_i)) \). As for the simple classifiers (section 3.2.1), we can set a real-valued limit \( \beta \in \mathbb{R} \), and define the classifier as:

\[ \hat{l}_i(\beta) = sgn(f_n(X_i) - \beta) \quad (20) \]

If \( \beta > 0 \), then the number of false positives \( (l_i = -1 \text{ and } \hat{l}_i = 1) \) is smaller, but the number of false negatives \( (l_i = 1 \text{ and } \hat{l}_i = -1) \) is bigger, relative to the \( \beta = 0 \) case, and the opposite is true if we choose \( \beta < 0 \). Taking all possible values for \( \beta \in \mathbb{R} \) defines the receiver operating characteristic curve, or ROC curve. The area under the ROC curve is a common measure for the quality of a binary classifier. The classifier is perfect if there exists a value of \( \beta \) such that all estimated labels are equal to the true labels; in this case the area
under the curve is equal to 1. Figure 5 shows one example of active learning, with ROC curves corresponding to different numbers of labeled signals. As expected, the classifier improves on average when the labeled set gets bigger; and the active learner becomes better than the simple PGA classifier as soon as $n \geq 10$.

Figure 5: ROC curves for the PGA classifier (black) and for 6 active learners after $n$ simulations ($n = 5, 10, 20, 50, 100$ and $200$).

Another metric can be used to measure performance: the precision/recall breakeven point [42]. Precision is the percentage of samples a classifier labels as positive that are really positive. Recall is the percentage of positive samples that are labeled as positive by the classifier. By altering the decision threshold on the SVM we can trade precision for recall, until both are equal, therefore defining the precision/recall breakeven point. In this case the number of false positives and false negatives are equal (see figure 4). Let us denote by $N_+$ the number of instances where the displacement $Z$ is greater than the threshold (on a total of $N$ signals in the database):

$$N_+ = \# \{ i \in [1, N] \mid l_i = 1 \} . \quad (21)$$

We sort all instances according to their score, i.e. we find a permutation $\sigma$ such that $f_n(X_{\sigma(1)}) \leq \cdots \leq f_n(X_{\sigma(N)})$. Then the precision/recall breakeven point (PRBP) is equal to the proportion of positive instances among the $N_+$ instances with the highest score:

$$\text{PRBP} = \frac{\# \{ i \in [1, N] \mid l_i = 1 \text{ and } \sigma(i) > N - N_+ \}}{N_+} . \quad (22)$$
This criteria does not depend on the number of true negatives (unlike the false positive rate, used in the ROC curve). In particular, it is not affected by our choice of preprocessing of the training data, where we discarded all the weak signals ($L_i < Y$). Both metrics are affected by our choice to discard the very strong signals ($L_i > 6Y$), but the effect is negligible in both cases.

3.3. Binary classification results for simple elasto-plastic structures

Considering the simple elasto-plastic structure presented in section 2.2, performances of different classifiers are compared considering the precision / recall breakeven point, in order to highlight the effectiveness of the active learning algorithm. More precisely, we compare different orderings of all signals, since only the order matters to the PRBP; for instance, the PGA does not give directly a label, but we can compute the PRBP of the PGA classifier with equation 22 using the permutation $\sigma_{PGA}$ that sorts the PGA of all signals. Thus, we compare:

1. the simple PGA and maximum linear displacement classifiers $\hat{I}_{PGA}$ and $\hat{I}_L$, defined in section 3.2.1. These simple classifiers are defined with the $N = 33718$ signals and labels;
2. neural networks, trained with all instances and all labels (ie, with the $N$ signals and labels), with either all 13 parameters, or just 4 of them: $(L, PGA, V, \omega_0)$ (the linear displacement, peak ground acceleration, peak ground velocity and filter frequency, see section 3.3.3 for justification of this choice). The neural networks we used are full-connected Multi Layered Perceptrons (MLPs) with 2 layers of 26 and 40 neurons for $X \in \mathbb{R}^4$, and two layers of 50 and 64 neurons for $X \in \mathbb{R}^{13}$. This classifier is considered in this study for completeness to show the best results we have achieved with the seismic indicators considered in $\mathbb{R}^4$ and in $\mathbb{R}^{13}$ with all their $N$ instances;
3. SVMs given by our active learning methods.

In the active learning category, the performance depends on the number $n$ of simulations (between $n = 10$ and $n = 1000$). So, the results shown in this section are functions of the number $n$ of labeled training instances, in logarithmic scale. For completeness, the evaluation of the performances first concerns the structure with its nominal parameters at 5 Hz (section 3.3.1) and secondly, structures at 2.5 Hz and 10 Hz (section 3.3.2).
3.3.1. Results for the nominal settings

Figure 6 shows performances of 3 active learning classifiers over 20 test cases considering nominal settings presented in section 2.2. Their performances are compared with those of 4 classifiers built using all the $N$ signals. The first two classifiers are two single-input classifiers respectively based on the PGA and the Linear displacement and the two others are neural networks respectively based on the inputs $X \in \mathbb{R}^4$ and $X \in \mathbb{R}^{13}$. These 4 classifiers are represented as horizontal lines, since they do not depend on $n$. Moreover, this figure shows the PRBP of (i) a linear SVM using all 13 parameters (in blue), (ii) a linear SVM using only 4 parameters $(L, PGA, V, \omega_0)$ (in red) and (iii) a radial basis function (RBF) SVM, using the same 4 parameters (in yellow). As active learners depend on the choice of the first two samples, results of figure 6a are obtained choosing 20 pairs of starting points $(j_1, j_2)$, then averaging the performance, knowing that the same starting points were used for all three types of SVMs. For completeness, figure 6b shows the worst and best performances of the 3 classifiers on the 20 test cases.
Figure 6: Performances of 3 active learning classifiers over 20 test cases. (a) Average. (b) Worst and best.

Figure 6a shows that active learning gives a much better classifier than the standard practice of using a single parameter (usually the PGA). The linear SVM with only 4 variables has initially the best performance on average, up to \( n = 150/200 \) simulations. The full linear SVM with 13 variables is better when the number of simulations is at least 200. The RBF kernel in \( \mathbb{R}^4 \) appears to have the best performance with \( n = 1000 \) labeled instances, outperforming the neural network in \( \mathbb{R}^4 \) using all \( N \) instances; however, it has a higher variability, as can be seen in figure 6b. Radial basis function SVMs with 13 parameters seem to always perform very poorly, and are not represented here. Figure 6b shows the lowest and highest score of all 20
test cases, independently for each number of simulations (one active learner can perform poorly at some point, and much better later, or the other way around). So, in conclusion, (i) active learners need a minimum of \( n = 30 - 40 \) simulations, otherwise they can end up worse than using the simpler PGA classifier, (ii) between \( n = 50 \) and \( n = 200 \) simulations, the linear SVM in \( \mathbb{R}^4 \) is the best choice, and has a relatively small variability and (iii) the RBF kernel seems quite unpredictable for less than \( n = 1000 \) simulations, and its performance depends wildly on the starting points, probably because of over-fitting.

3.3.2. Results for different settings

Our methodology is very general and can be applied to a variety of structures. As an example, we compared the same classifiers on two structures with two different main frequencies, 2.5 Hz and 10 Hz, instead of the original 5 Hz. The elasticity limit \( Y \) was also changed so that approximately one third of all signals result in inelastic displacement: \( Y = 9 \cdot 10^{-3} \) m for the 2.5 Hz structure, \( Y = 5 \cdot 10^{-3} \) m for 5 Hz and \( Y = 1 \cdot 10^{-3} \) m for 10 Hz. The failure threshold was always chosen as \( 2Y \), which resulted in about 8.8\% of all signals attaining it for the 2.5 and 10 Hz cases, compared to 5.7\% in the 5 Hz setting. As shown in figure 7, the performances of active learners are very similar to the 5 Hz case, and the same conclusions apply. The performances of classifiers based on a single parameter, on the other hand, can vary a lot depending on the frequency of the structure: the PGA classifier provides a good classifier at high frequency (PRBP= 0.797 at 10 Hz) but performs poorly at low frequency (PRPB=0.6 at 2.5 Hz, it does not appear in figure 7); while the linear displacement does the opposite (PRBP= 0.798 at 2.5 Hz, but PRBP= 0.69 at 10 Hz). These results show that the active learning methodology is not just more precise, but also more flexible than the simple classifiers, and that with just 50 to 200 simulations it approaches the performance of a neural network using 33718 simulations.
3.3.3. Remark about the dimension reduction

In the linear case the score is equal to the distance to the hyperplane: \( f_n(X) = W^T X + c \) (see equation 15). Therefore, since all 13 components were standardized in the preprocessing procedure (see section 3.1), we can see which of the components of \( X \) are the most important for the classification simply by looking at the values of the components of \( W \). Figure 8 shows that the values of \( W \) are roughly the same for all 20 test cases. After \( n = 1000 \) simulations, the coefficients for the PGA and maximum linear displacement \( L \) end up between 3 and 4, the value for the maximum velocity \( V \) is around 1, and the value for the signal main frequency \( \omega_0 \) is around \(-1\). The other 9 components of \( W \) (when working with \( X \in \mathbb{R}^{13} \)) are all between \(-1\) and 1, but are smaller (in absolute value) than these 4 components. As can be seen in the previous sections, reducing the dimension from 13 to 4 allows for a faster convergence, although the converged classifier is usually less precise. Continuing the active learning after \( n = 1000 \) simulations changes only marginally the results; even with \( X \in \mathbb{R}^{13} \), both the PRBP and the values of \( W \) stay roughly the same between \( n = 1000 \) and \( n = 5000 \) simulations.
3.4. Synthesis and Recommendations

Although the results of the comparative study are not shown here for the sake of brevity, a simple but crucial preprocessing of the data is first necessary to improve the performances of the SVM classifiers especially when they are linear. Moreover, as expected, the results presented the section 3.3 show that the performances of the SVM classifiers are structure-dependent. As well in $\mathbb{R}^4$ as in $\mathbb{R}^{13}$ the input parameters are indeed more or less well correlated with the output according to the structure considered. The proposed methodology is nevertheless very general and can be applied to a variety of structures. Thus, these results show that a minimum of $n = 100$ simulations, selected via an active learning algorithm (sections 3.2.3 and 3.2.4), are necessary in order to obtain very precise classifiers. For $n = 100$ simulations, it is shown that linear SVM classifiers in $\mathbb{R}^4$ are sufficient. In this case, for the class of structures (elasto-plastic structures), the seismic scenario and the type of GMSM considered, the main seismic IM parameters are the classical parameters: $PGA$, $V$, $L$, and $\omega_0$. We will see also in the next section that such a classifier allows good estimations of fragility curves.

4. Fragility curves

As mentioned in the introduction, different procedures can be used to construct non parametric fragility curves (see e.g. [38], [19]). Here, we assume that they are simply constructed based on k-means clustering of the IM data. In a Monte Carlo-based approach this means that in each cluster, the empirical probability of failure is evaluated by the ratio between the
number of structural responses that exceed the limit threshold and the cluster size, that is, the number of structural responses belonging to the cluster. With SVM classifiers which give to each signal $i$ a real-valued score $f_n(X_i)$ whose sign expresses the estimated label, we first need to assign a probability to estimate fragility curves. In this section we explain how SVMs can be used to estimate fragility curves, using either the score functions (which can be viewed as optimal seismic IMs since a perfect classifier would lead to a fragility curve in the form of a unit step function when the problem is linearly separable) or classical IM indicators such as the PGA or the spectral acceleration (which is replaced here by the linear displacement $L$).

4.1. Proabilistic interpretation of the output of the SVM

The probabilistic interpretation of the output of the SVM depends only on the score $f_n(X)$. For a perfect classifier, the probability would be 0 if $f_n(X) < 0$ and 1 if $f_n(X) > 0$; for our SVM classifiers we use a logistic function:

$$p_n(X) = \frac{1}{1 + e^{-af_n(X)+b}}, \quad (23)$$

where $a$ and $b$ are the slope and intercept parameters of the logistic function ($b$ should be close to 0 if the classifier has no bias, giving a probability of 1/2 to signals with $f_n(X) \approx 0$). These parameters $(a, b)$ are estimated by maximizing the likelihood function from the logistic function (equation 23) on the labeled set $\{(X_{j1}, l_{j1}), \ldots, (X_{jn}, l_{jn})\}$.

4.2. Fragility curves estimations

The estimation of the score-based fragility curve is simply given by equation 23. However, if we are interested by the PGA-based fragility curve (or an other IM-based fragility curve) the classifier has to be used first to predict the scores and the associated probabilities (given by equation 23) of several new input parameters. These new parameters correspond to those which have not been selected for the definition of the classifier or others generated from new simulations of the GMSM. Then, as in a Monte Carlo-based approach, k-means clustering has to be used on the IM indicator of interest. In each cluster, the probability of failure is then evaluated by averaging the probabilities associated to the input parameters belonging to the cluster.
4.2.1. Score-based fragility curve

To compare the estimation given by equation 23 with the empirical failure probability of signals with a given score, we divide the set of indices \{1, ... , N\} of our database \(\mathbf{X}\) into \(K\) groups \((I_1, ..., I_K)\) depending on their score \(f_n(\mathbf{X}_i)\), with the k-means algorithm; then we define the estimated and reference probabilities of each group:

\[
\begin{align*}
p_{k}^{\text{est}} &= \frac{1}{n_k} \sum_{i \in I_k} p_n(X_i), \\
p_{k}^{\text{ref}} &= \frac{1}{n_k} \# \{ i \in I_k | l_i = 1 \}, \quad \text{with} \quad n_k = \#I_k. \tag{24}
\end{align*}
\]

We can now compute the discrete \(L_2\) distance between these two probabilities:

\[
\Delta_{L_2} = \sqrt{\frac{1}{N} \sum_{k=1}^{K} n_k (p_{k}^{\text{ref}} - p_{k}^{\text{est}})^2}, \tag{25}
\]

with \(N = \sum_{k=1}^{K} n_k\).

Figure 9 shows this distance for different classifiers using \(n = 20, 50, 100, 200, 500\) and \(1000\) labeled instances. The three classifiers (linear SVM in \(\mathbb{R}^{13}\), linear SVM in \(\mathbb{R}^{4}\), and RBF kernels in \(\mathbb{R}^{4}\)) are compared on 20 test cases, using 20 pairs of starting points (the same for all three). The solid lines show the average \(L_2\) errors, and the dashed lines show the minimum and maximum errors among all test cases. The average error goes down from 15% after \(n = 20\) simulations to less than 3% after \(n = 1000\) simulations for the linear SVM in \(\mathbb{R}^{13}\), and from 9% to less than 2% for the linear SVM in \(\mathbb{R}^{4}\). For the SVM using RBF kernels (in yellow in figure 9), the average error does not decrease as the number of simulations increases, and ends up around 20% after \(n = 1000\) simulations.
Figure 9: Distance between the reference and estimated fragility curves for 3 different active learners.

Figure 10 shows typical examples of fragility curves obtained with each method after \( n = 100 \) and \( n = 1000 \) simulations. Recall that the logistic functions (in red) are not fitted using all the real data (in blue), but only the labeled set, i.e. \( n = 100 \) or \( n = 1000 \) signals. The linear SVM in \( \mathbb{R}^4 \) has the least errors in terms of probabilities, although its PRBP is smaller than the linear SVM in \( \mathbb{R}^{13} \) when using \( n = 1000 \) labeled instances.
Figure 10: Reference and estimated fragility curves using (a and d) a linear SVM in $\mathbb{R}^{13}$, (b and e) a linear SVM in $\mathbb{R}^{4}$ and (c and f) a RBF SVM in $\mathbb{R}^{4}$, with (a, b and c) $n = 100$ or (d, e and f) $n = 1000$ labeled instances.
The radial basis function kernel shows a very "strange" behaviour. The probability of failure is not even an increasing function of the score (figures 10c and 10f); in particular, signals with a very negative score still have a 5—10% chance of exceeding the threshold. This strange shape of \( p_{\text{ref}}^k \) explains why the \( \Delta_{L_2} \) error of RBF kernels is so high (figure 9), since we tried to fit a logistic curve on a non monotinous function. The reason for this major difference between linear and RBF kernels can be understood if we look at the maximum total displacement \( Z \) as a function of the score \( f_n(X) \), using both kernels (see figure 11). Let us keep in mind that the RBF classifier at \( n = 1000 \) simulations is the most precise of all our active learners; it has the fewest false positives and false negatives of all (see table 1). The sign of the RBF score is thus an excellent predictor for binary classification.

<table>
<thead>
<tr>
<th>linear kernel</th>
<th>( f(X) &lt; 0 )</th>
<th>( f(X) &gt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Z &gt; 2Y )</td>
<td>1020</td>
<td>4711</td>
</tr>
<tr>
<td>( Z &lt; 2Y )</td>
<td>27175</td>
<td>812</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RBF kernel</th>
<th>( f(X) &lt; 0 )</th>
<th>( f(X) &gt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Z &gt; 2Y )</td>
<td>1009</td>
<td>4722</td>
</tr>
<tr>
<td>( Z &lt; 2Y )</td>
<td>27287</td>
<td>700</td>
</tr>
</tbody>
</table>

Table 1: Confusion matrix after \( n = 1000 \) simulations for the linear SVM in \( \mathbb{R}^4 \) (left) and the RBF SVM in \( \mathbb{R}^4 \) (right).

Figure 11 shows that for the linear classifier, the score is a good predictor of the maximum total displacement \( Z \), with a monotonous relation between the two; therefore the probability that a \( Z > 2Y \) is well-approximated by a logistic function of the score. The RBF score, on the other hand, is a poor predictor of the probability of failure, since the relation between the score and the maximum total displacement \( Z \) is not monotonous. We can now understand the very high \( \Delta_{L_2} \) errors for RBF kernels. Looking at figure 11, we can see that the weakest signals (\( Z = 0.005 \), just above the elasticity limit) have a RBF score between \(-1\) and \(-0.4\). Since these weak signals are very common in our database, the reference probability \( p_{\text{ref}}^k \) goes rapidly from 0.5 for \( f_n^{\text{RBF}}(X) = 0 \) to almost 0 for \( f_n^{\text{RBF}}(X) = -0.5 \) (see figures 10c and 10f), not because the number of positive signals changes significantly between \( f_n^{\text{RBF}}(X) = -0.5 \) and \( f_n^{\text{RBF}}(X) = 0 \), but because the number of negative signals is more than 20 times bigger. The linear kernels do not have this problem, and therefore have much lower \( \Delta_{L_2} \) errors.
ROC curves (figure 12) give us another way to look at this dilemma between linear and RBF kernels. If we look at the unbiased (i.e. $\beta = 0$) classifiers, the RBF is slightly superior: it has fewer false positives and slightly fewer false negatives than the linear classifier. However, when we choose a negative limit $\beta$ (see equation 20), for example $\beta = -0.5$, then some of the weakest signals end up over the limit ($f_{1000}^{RBF}(X) > \beta$) and thus have an estimated label of $\hat{L}(\beta) = 1$. Since these weak signals are so common, the false positive rate becomes extremely high.

Figure 11: $Z$ as a function of the score given after $n = 1000$ simulations by (a) the linear SVM in $\mathbb{R}^4$ and (b) the RBF SVM in $\mathbb{R}^4$.

Figure 12: (a) ROC curves for two SVM classifiers using linear and RBF kernels, with specific values for the unbiased (i.e. $\beta = 0$) classifiers. (b) zoom on the upper-left corner.
4.2.2. PGA-based (resp. L-based) fragility curve

In the previous section we used the score $f_n(X)$ as the parameter on the x-axis to build the fragility curves. However, our method assigns a probability $p_n(X)$ to each signal, depending only on a few parameters. If we consider this probability as a function of 4 parameters ($p_n(L, V, PGA, \omega_0)$ if $X \in \mathbb{R}^4$), then we can use any of those parameters, the PGA for example, to define a posteriori a fragility curve depending on just this parameter, averaging over the other ones:

$$p_n(PGA) = \mathbb{E}[p_n(X)|PGA].$$

(26)

In practice, to obtain numerically the corresponding estimated and reference probabilities, we use k-means algorithm, and divide the database into $K$ groups ($I_1, ..., I_K$) depending on their PGA (resp. on $L$), instead of the score, then compute $p^e_k$ and $p^{ref}_k$ using equation 24. Figures 13 and 14 show two examples of such curves, using the PGA or the maximum linear displacement $L$; $p_n(X)$ is computed using a linear SVM classifier in $\mathbb{R}^4$ with $n = 100$ (left) or $n = 1000$ (right) simulations. In this case, we can show all 20 test cases in a single figure, since they share a common x-axis (which is not true when we used the score).

Figure 13: Reference and estimated fragility curves as a function of the PGA, using (a) $n = 100$ and (b) $n = 1000$ labeled points.
We now have a fully non parametric fragility curve. The distance between the reference and estimated curve is very small in both cases, even using just $n = 100$ labeled instances, although the spread is smaller when we add more data points.

4.3. Trading precision for steepness

The PGA-based and L-based fragility curves (figures 13 and 14) are very close to the reference curves; the distance $\Delta_{L^2}$ between reference and estimated curves is very small, even smaller than in the case of score-based fragility curves. In this case, why even bother with score-based fragility curves? What is their benefit, compared to easily-understandable, commonly accepted PGA-based curves?

The difference is in the steepness of the curve. Formally, when we construct a fragility curve, we choose a projection $F : \mathbb{R}^4 \mapsto \mathbb{R}$ to use as the x-axis. This projection $F(X)$ can be one of the 4 variables (for example the PGA), or the score $f_n(X)$, which can be a linear or nonlinear (in the case of RBF kernel) combination of the 4 variables. We then use the k-means algorithm to make groups of signals who are "close" according to this projection, i.e. signals with the same PGA or the same score; then we compute the estimated probability $p_k^{est}$ for each group. Let us assume for a while that our estimation is very precise, so that $p_k^{est} = p_k^{ref} \forall k$. In this case, which
fragility curve gives us the most information? To see this, we define:

\[ R^{(F)} = \frac{1}{N} \sum_{k=1}^{K} n_k \phi(p_{k}^{\text{est}(F)}). \]  

(27)

for some nonnegative-valued function \( \phi \). Intuitively, a perfect classifier would give each signal a probability of 0 or 1, while a classifier which assigns a probability of 1/2 to many signals is not very useful. Therefore, we want \( \phi \) to be positive on (0,1), equal to 0 for \( p = 0 \) and \( p = 1 \). If we choose:

\[ \phi(p) = -p \ln(p), \]  

(28)

then \( R^{(F)} \) can be seen as the entropy of the probability \( p(X) \), which would be equal to 0 for a perfect classifier and has higher values for a "useless" classifier. Another choice would be:

\[ \phi(p) = \mathbb{1}_{p \in [0,0.9]}. \]  

(29)

In this case, \( R^{(F)} \) also has a clear physical meaning: it is the proportion of "uncertain" signals, i.e. signals such that \( p_{k}^{\text{est}}(X) \in [0.1, 0.9] \). Table 2 shows the value of \( R^{(F)} \), using the entropy version, for different choices of projection (score, PGA, or linear displacement). We can see on this table that the PGA-and L-based fragility curves are extremely precise, with very low values of \( \Delta L_2 \) (this can also be seen in figures 13 and 14), but their entropy is much higher than the score-based fragility curves.

<table>
<thead>
<tr>
<th>( n ) ( = 100 )</th>
<th>( \Delta L_2 ) (%)</th>
<th>score</th>
<th>PGA</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>entropy (( 10^{-2} ))</td>
<td>3.8 ± 1.6</td>
<td>2.6 ± 1</td>
<td>2.8 ± 0.9</td>
<td></td>
</tr>
<tr>
<td>( n = 1000 )</td>
<td>( \Delta L_2 ) (%)</td>
<td>1.7 ± 0.6</td>
<td>1.6 ± 0.3</td>
<td>1.4 ± 0.4</td>
</tr>
<tr>
<td>entropy (( 10^{-2} ))</td>
<td>7.2 ± 1.2</td>
<td>13.3 ± 1.5</td>
<td>13.6 ± 1</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Precision and entropy of fragility curves using different projections (average and standard deviation over 20 test cases), for \( n = 100 \) (top) or \( n = 1000 \) (bottom) labeled points.

One surprising fact of table 2 is that the entropy is smaller at \( n = 100 \) compared to \( n = 1000 \) in all three cases. This shows that after only \( n = 100 \) mechanical calculations, all our classifiers tend to slightly overestimate the
steepness, and give fragility curves that are actually steeper than the reality (and also steeper than the more realistic $n = 1000$ curves). This was also seen in figures 10b and 10e: at $n = 100$ simulations the estimated curve is steeper than the reference curve, which gives an estimated entropy smaller than the reference entropy. Using the other choice of $\phi$ gives the same conclusions: the proportion of signals with $p_k^{\text{est}} \in [0.1, 0.9]$ is 18.2% if we use the score, but it is around 28% for both the PGA and maximum linear displacement. Therefore, the choice of the projection used for a fragility curve is a trade between precision and steepness. Keep in mind that the values of the entropy for different choices of projection can be obtained after the active learning, and the computational cost is very small (mostly the cost of k-means). As a consequence, this choice can be made a posteriori, from the probabilities assigned to each signal.

4.4. Remark about the specificity of active learning

Using the score to compute the probabilities (equation 23) on the whole dataset $\mathbf{X}$ is absolutely mandatory, even if one is only interested in the PGA fragility curves. In particular, looking only at the labeled set $\mathcal{L}$ to find directly a probability of failure depending on the PGA gives extremely wrong results. Figure 15 shows not only the reference and estimated fragility curves previously defined, but also the $n = 1000$ points of the labeled set $\mathcal{L}$ and an empirical probability built from it.

![Figure 15: Empirical fragility curve using only the labeled set (violet).](image)

This empirical fragility curve was computed by using the k-means algorithm on the PGA values of the labeled set $\mathcal{L}$, then taking the empirical failure probability in each group. The result looks like... a constant around
Looking only at \( L \), the PGA does not even look correlated with failure. The reason for this surprising (and completely wrong) result is the active learning algorithm. \( L \) is not at all a random subset of \( \mathcal{X} \); it is a carefully chosen set of the data points with maximal uncertainty, which means that most of them end up very close to the final hyperplane (and have a final score very close to 0), but also that they span as much of this hyperplane as possible. When these points are projected on any axis not perpendicular to the hyperplane (that is, any axis but the score), for example the PGA (which is one of the components of \( \mathcal{X} \)), the empirical probability is roughly equal to the constant \( 1/2 \), which is not representative at all of the underlying probability. Using equation 23 (then eventually fitting the results to a lognormal curve for the PGA) is the right way to go.

Finally, these results also illustrate that the data points obtained thanks to the active learning algorithm can not be used for a cross-validation procedure. So, if we want to validate the results in a rigorous way, it is necessary to use a separate testing set following for example the procedure described in the reference [42].

5. Conclusion

This paper proposed an efficient methodology for estimating non parametric seismic fragility curves by active learning with a Support Vector Machines classifier. We have introduced and studied this methodology when aleatory uncertainties have a predominant contribution in the variability of structural response, that is to say when the contribution of uncertainties regarding seismic excitation is "much larger" than the contribution of uncertainties regarding structural capacity. In this work, structure was considered as deterministic. In this framework, a perfect classifier, if it exists, would lead to a fragility curve in the form of a unit step function, i.e. corresponding to a fragility curve "without uncertainty". That means the output of this classifier, which is a real-valued score, would be the best seismic IM indicator to evaluate the damaging potential of the seismic signals, knowing that such a classifier would necessary be both structure and failure criterion-dependent, with possibly a dependence on the ground motion characteristics (near-fault type like, broadband, etc).

The proposed methodology makes it possible to build such a (non perfect) classifier. It consists in (i) reducing input excitation to some relevant parameters and, given these parameters, (ii) using a SVM for a binary classification
of the structural responses relative to a limit threshold of exceedance. Selection of the mechanical numerical calculations by active learning dramatically reduces the computational cost of construction of the classifier. The output of the classifier, the score, is the desired IM indicator which is then interpreted in a probabilistic way to estimate fragility curves as score functions or as functions of classical seismic IMs.

This work shows that a simple but crucial preprocessing of the data (i.e. Box-Cox transformation of the input parameters) makes it possible to use a simple linear SVM to obtain a very precise classifier after just one hundred simulations, that is to say with one hundred mechanical calculations. Moreover, this work shows that the parameters of the GMSM can be used additionally with the classical IM parameters to build the classifier and to improve its performance. Thus, for the class of structures considered, with only four classical seismic parameters ($PGA, V, L, \omega_0$), the score-based fragility curve is very close to the reference curve (obtained with a massive Monte Carlo-based approach) and steeper than the PGA-based one, as expected. L-based fragility curves appear to perform about as well as PGA-based ones in our setting. Advanced SVMs using RBF kernel result in less classification errors when using one thousand mechanical calculations, but does not appear well suited to making fragility curves.

A naive way to take into account epistemic uncertainties would consist in building a classifier for each set of structural parameters. Nevertheless, such a method would not be numerically efficient. Another way could be to assume that epistemic uncertainties have small influence on the classifier evaluated, for example, for the median capacity of the structure. Thus, only calculations of linear displacements would be necessary to estimate the corresponding fragility curve. However, to avoid such assumptions, some research efforts have to be devoted to propose an efficient overall methodology that takes into account the two types of uncertainties.

6. Acknowledgement

This research was supported by the French National Research Agency (grant ANR CHORUS MONU-0005), by CEA (French Alternative Energies and Atomic Energy Commission) and SEISM Institute (www.institut-seism.fr/en/).
Appendix A. Parameter identification of the GMSM and simulation

Appendix A.1. Modulation parameters

For a target recorded accelerogram \( a(t) \), we determine the modulation parameters \( \alpha \) by matching the cumulative energy of the accelerogram
\[
E_a(t) = \int_0^t a^2(\tau) d\tau
\]
with the expected cumulative energy\( E_s(t) = \mathbb{E} \left[ \int_0^t s^2(\tau) d\tau \right] = \int_0^t q^2(\tau, \alpha) d\tau \) of the stochastic process \( s(t) \), which does not depend on the filter parameters, if the high-pass post-processing is neglected. Thanks to the definition of \( \sigma_f(t) \), this expected energy only depends on the modulation parameters \( \alpha \). To match the two cumulative energy terms, we minimize the integrated squared difference between them:
\[
\hat{\alpha} = \arg\min \alpha \int_0^T [E_s(t) - E_a(t)]^2 dt
= \arg\min \alpha \int_0^T \left[ \int_0^t q^2(\tau, \alpha) d\tau - \int_0^t a^2(\tau) d\tau \right]^2 dt. \tag{A.1}
\]

This minimization is done with the Matlab function \texttt{fminunc}. Note that the PGA of the generated signal is not necessarily equal to that of the recorded one on average.

Appendix A.2. Filter parameters

For the filter parameters \( \lambda = (\omega_0, \omega_n, \zeta_f) \), we use the zero-level up-crossings, and the positive minima and negative maxima of the simulated signal \( s(t) \) and target signal \( a(t) \). These quantities do not depend on scaling, thus we use only the un-modulated process
\[
y(t) = \int_{-\infty}^t \frac{h[t - \tau, \lambda(\tau)]}{\sigma_f(t)} w(\tau) d\tau. \tag{A.2}
\]

For a given damping ratio \( \zeta_f \), we can identify the frequencies \( (\omega_0, \omega_n) \) by matching the cumulative count \( N_a(t) \) of zero-level up-crossings of the target signal \( a(t) \) with the same expected cumulative count \( N_x(t) \) for the simulated signal, given by:
\[
N_x(t) = \int_0^t \nu(\tau) r(\tau) d\tau, \tag{A.3}
\]
where \( \nu(\tau) \) is the mean zero-level up-crossing rate of the process \( y(t) \) and \( r(\tau) \) is an adjustment factor due to discretization (usually between 0.75 and
1). Since \( y(t) \) is a Gaussian process with zero mean and unit variance, the mean rate \( \nu(\tau) \), after simplification, is given by:

\[
\nu(t) = \frac{\sigma_{\dot{y}}(t)}{2\pi},
\]

where \( \sigma_{\dot{y}}(t) \) is the standard deviation of the time derivative \( \dot{y}(t) \) of the process:

\[
\sigma_{\dot{y}}(t)^2 = \frac{1}{\sigma_f(t)^2} \int_{-\infty}^{t} \left[ \dot{h}(t - \tau, \lambda(\tau)) - h(t - \tau, \lambda(\tau)) \right]^2 d\tau,
\]

assuming we neglect integrals over a fraction of a time step in the discretization.

To identify the damping ratio \( \zeta_f \), we use the cumulative count of positive minima and negative maxima. Indeed, in a narrow-band process (\( \zeta_f \) close to 0), almost all maxima are positive and almost all minima are negative, but the rate increases with increasing bandwidth (larger \( \zeta_f \)). An explicit formulation exists for this rate but it involves computing the second derivative of \( y(t) \) [43], thus it is easier to use a simulation approach, by counting and averaging the negative maxima and positive minima in a sample of simulated realizations of the process, then choosing the value that minimizes the difference between real and simulated cumulative counts.

Appendix A.3. Definition of the estimator \( p_{KDE}(\theta) \)

Let \( (\theta_1, \theta_2, \ldots, \theta_{N_r}) \) be a multivariate independent and identically distributed sample drawn from some distribution with an unknown probability density function \( p(\theta), \theta \in \mathbb{R}^d \). The kernel density estimator \( p_{KDE} \) of \( p(\theta) \) is:

\[
p_{KDE}(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \phi_{\mathbf{H}}(\theta - \theta_i),
\]

where \( \phi_{\mathbf{H}} \) is a Gaussian kernel centered at 0 with covariance matrix \( \mathbf{H} \). A classical measure of closeness of the estimator \( p_{KDE}(\theta) \) to the unknown density \( p(\theta) \) is the asymptotic mean integrated squared error (AMISE), defined as:

\[
\text{AMISE}(\mathbf{H}) = (4\pi)^{-d/2}|\mathbf{H}|^{-1/2}N_r^{-1} + \frac{1}{4} d^2 \int \text{Tr}^2 \left[ \mathbf{H} \text{Hess}_p(\theta) \right] d\theta,
\]

A.3
where $\text{Tr}$ is the trace operator and $\text{Hess}_p$ is the Hessian of $p$. If we write $H = \beta^2 F$ with $\beta \in \mathbb{R}$ and we suppose that $F$ is known, then the AMISE is minimized for:

$$\beta_{\text{opt}} = \left[d(4\pi)^{d/2} N_r R(p, F)\right]^{-\frac{1}{d+4}},$$  \hspace{1cm} (A.8)

where $R$ still depends on the underlying (and unknown) distribution $p(\theta)$:

$$R(p, F) = \int \text{Tr}^2 [F \text{Hess}_p(\theta)] d\theta.$$  \hspace{1cm} (A.9)

Following Kristan [41], $F$ can be approximated by the empirical covariance matrix $\hat{\Sigma}^{\text{smp}}$ of the observed samples and $R(p, F)$ can be approximated by

$$\hat{R} = \left(\frac{4}{(d+2)} N_r\right)^{-\frac{1}{d+4}} \sum_{i,j=1}^{N_r} \phi_G (\theta_i - \theta_j) \left(\frac{2}{N_r} (1 - 2m_{ij}) + (1 - m_{ij})^2\right),$$  \hspace{1cm} (A.10)

where

$$m_{ij} = (\theta_i - \theta_j)^T \hat{G}^{-1} (\theta_i - \theta_j), \quad \hat{G} = \left(\frac{4}{(d+2)} N_r\right)^{-\frac{2}{d+4}} \hat{\Sigma}^{\text{smp}}.$$  \hspace{1cm} (A.11)

The estimator $p_{KDE}(\theta) = \frac{1}{N_r} \sum_1^{N_r} \phi_H (\theta - \theta_i)$ is now fully defined.

References


[34] N. Simon Kwong and Anil K. Chopra. Evaluation of the exact conditional spectrum and generalized conditional intensity measure methods


